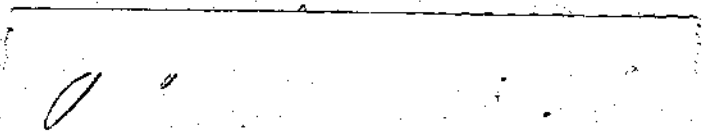
In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.



ON THE OPTIMAL CONTROL OF DISCRETE SYSTEMS WITH BOUNDED STATE VARIABLES AND BOUNDED CONTROL VARIABLES

A THESIS

Presented to

The Faculty of the Graduate Division

by

Joseph Samuel Boland, III

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Electrical Engineering

Georgia Institute of Technology

August, 1968

ON THE OPTIMAL CONTROL OF DISCRETE SYSTEMS

WITH BOUNDED STATE VARIABLES

AND BOUNDED CONTROL VARIABLES

Approved:

Chairman

Date approved by Chairman: August 26, 1968

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to Dr. Roger P. Webb, my thesis advisor, for his guidance and assistance during the development of this dissertation. Appreciation is also extended to Drs. J. R. Rowland and S. L. Dickerson for their time and services as members of the reading committee. In addition, I wish to acknowledge the assistance of Dr. J. H. Schlag, who suggested the problem and made many helpful comments and suggestions throughout the course of this work.

Special appreciation is expressed to the Electrical Engineering Department and the National Aeronautical and Space Administration for their financial support during the period this research was conducted.

Finally, to my wife, Rowena, goes my deepest appreciation for her understanding and patience. Without her support and encouragement, this dissertation would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Figure Page

SUMMARY

In many control theory problems it is necessary to consider the physical or imposed bounds on the control variables and state variables of the system. The computation of the optimum control for such a system usually involves the solution of a two-point boundary value problem, which is a formidable task even with present-day computers. This dissertation is concerned with the problem of a linear discrete-input system with a quadratic cost functional, a convex constraint on the state variables, and a magnitude constraint on the control variables. The emphasis is on the development of a control algorithm that can be used to optimally control this system in real time.

The convex constraint on the state variables defines a region in the state space in which the trajectory must not be allowed. With a magnitude constraint on the control variables, there may be additional regions in the state space from which a trajectory cannot be prevented from entering the original unallowable region with the control available. A method for computing these additional unallowable regions is presented. Two theorems are presented that establish the optimal method of bounding the trajectory to prevent it from entering the unallowable regions.

An optimization algorithm is presented which includes three distinct procedures. The first is designed to compute the optimum unconstrained controls. This procedure uses ordinary calculus and is

used when both the controls and states remain within their prescribed bounds. The second procedure, which uses a coordinatewise gradient technique, is designed to compute the optimum magnitude constrained controls when the state boundary is not violated. The third procedure is designed to compute the optimum controls when both the control and state bounds have to be considered.

The applicability of the optimization algorithm is illustrated through two example problems. Several comparisons between results presented in this dissertation and existing results are given. The feasibility of using the optimization algorithm in an on-line controller is discussed. The advantages and limitations of the proposed method over existing methods are presented.

CHAPTER I

INTRODUCTION

The objective of this dissertation is to develop a method for controlling a discrete-input system subject to control saturation and state variable constraints such that a specified cost functional is minimized. A control algorithm that meets the above requirements is said to control the system in an optimal manner. The major contribution of this work is the development of a control algorithm that computes the optimal controls for systems with control and state variable constraints, is capable of operating on-line, and is applicable to systems of any order.

All control systems have limitations that cannot or should not be exceeded on both the controls and states of the system. A physical limitation, such as the maximum thrust of an aircraft engine is an example of a constraint that cannot be exceeded. A controller designed without consideration of such constraints will result in a non-optimal solution if the computed control exceeds the saturation value. Examples of constraints that should not be violated are the maximum allowable acceleration of a manned space vehicle and the maximum torque that the drive shaft of an automobile can withstand without structural failure. A controller designed without consideration of this type of limitation may cause damage to the system as well as yield a non-optimal solution.

In the design of controllers for bounded systems, there are four distinct problems to be considered: problems where no constraints need be considered; problems where only control constraints need be considered; problems where only state constraints need be considered; and the most general problem where both control and state constraints need be considered. The purpose of this dissertation is to present a method of controlling in an optimal manner systems in the last class.

History of the Problem

In the past 15 years the emphasis of work in system control has been on the optimal control of multidimensional plants. The demand for control of such complex systems as space vehicles, nuclear reactor plants, and chemical processing plants has been the prime motivation for the development of modern control theory. The early work dealt with the control of continuous systems to meet certain specifications, such as minimum time, minimum fuel, and minimum mean-square error. The previously well-established theory of the calculus of variations was used extensively in solving these problems. Later, other methods were introduced, such as Pontryagin's maximum principle (1), Bellman's dynamic programming (2), and many direct methods (3). Bounds on the control variables and state variables were handled by using the theory of Lagrange multipliers (4).

In the past six or seven years, primarily because of the rapid advancement of digital computer technology, a great deal of work has been done on the optimal control of discrete-input systems. The first work was the solution of minimal-time problems with quadratic cost

indices with no constraints on the controls and no bounds on the states of the system. Later this work was extended to include systems with constraints on the control variables. Desoer and Wing (5) solved the minimum-time regulator problem for linear systems with constraints on the magnitude of the control variables by introducing the concept of reachable regions. LeMay (6) extended this concept to include maximum regions of recoverability. Nagata, Kodama, and Kumagai (7) further extended the work to include a bound on one of the states of the system. Sarachik and Kranc (8) also extended the work of Desoer and Wing to systems with multiple controls and multiple outputs where each control may be subject to a different type of constraint.

Another class of problems that has received considerable attention in the last three or four years is the system with bounds on the state variables but with no bounds on the control variables. The earlier work in this area by Dreyfus (9), Berkovitz (10), and Rekasius (11) considered only continuous-input problems which require the solution of a two-point boundary value problem. The adjoint penalty function method of Fiacco and McCormick (12) was used in references (13) and (14) to obtain a suboptimal solution that can be made as accurate as desired. Schlag (15), using some of Kishi's results (16), developed an on-line controller for a discrete system that is optimal when considering a fixed number of steps into the future only.

The class of discrete-input problems with bounds on the state variables and on the control variables has received very little attention. Chung (17) obtained a suboptimal solution to this problem with

a quadratic cost functional using the adjoint penalty function method. His method, however, requires the solution of a two-point boundary value problem which requires considerable computation time. No method has been reported which yields a controller capable of computing the optimal controls in real time.

Statement of the Problem

A mathematical statement of the general problem considered in this dissertation is formulated in this section. The continuous problem is first formulated to form a basis for the formulation of the discrete problem. Several comparisons of the continuous and discrete problems are given in the following chapters.

The continuous problem with control and state constraints can be formulated with the aid of Figure 1. The plant to be controlled is an analog system with n state variables $\underline{x}(t)$ and m control variables $\underline{u}(t)$, where $m < n$. The plant can be described by n first order equations of the form

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{1.1}$$

where $\underline{x}(t)$ is an nth order vector, $\underline{u}(t)$ is an mth order vector and $\underline{f}$ is an nth order vector function. The problem can be formulated as a tracking problem as in Figure 1 or as a regulator problem in which $\underline{r}(t) \equiv \underline{0}$.

The function of the controller in Figure 1 is to compute the set of m control variables $\underline{u}(t)$ such that the system performs in an optimal

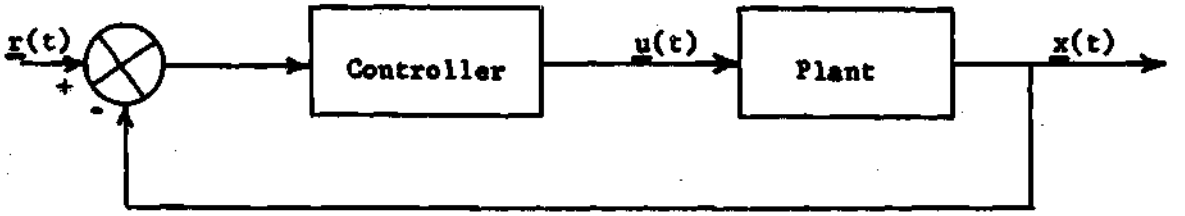


Figure 1. Block Diagram for Continuous System.

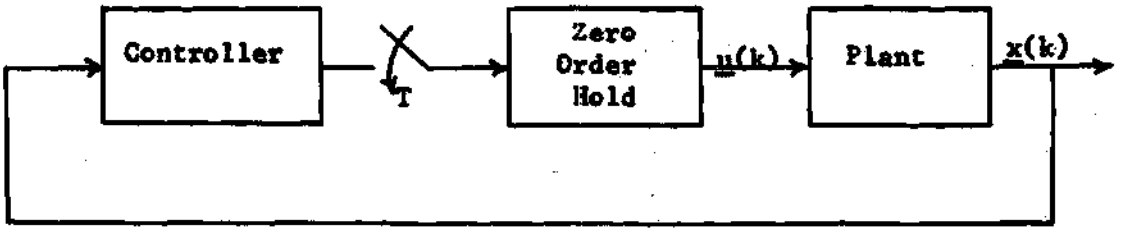


Figure 2. Block Diagram for Discrete System.

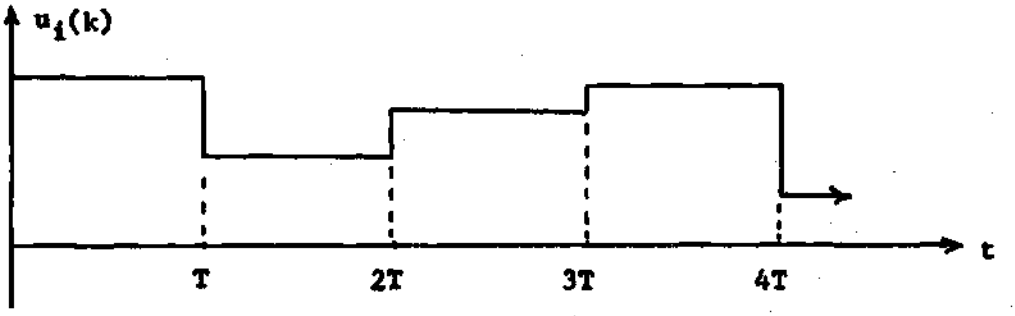


Figure 3. Example of i^{th} Component of Discrete Control.

manner. The criterion of optimality is usually a cost index which must be minimized. The cost index for continuous systems is an integral functional J which maps the controls and the trajectory of the system states into a real number. This functional can be expressed mathematically as follows:

$$J = \int_{t_o}^{T} g[\underline{x}(t), \underline{u}(t), t]dt \tag{1.2}$$

The constraint on the states of the system geometrically is an n dimensional hypersurface in the state space which must not be violated by the trajectory of the system. This n dimensional hypersurface can be expressed mathematically as a function $\Gamma[\underline{x}]$ which is identically equal to zero. If the system trajectory is to remain inside the boundary in the state space, the trajectory must satisfy the inequality

$$\Gamma[\underline{x}(t)] \leq 0 \tag{1.3}$$

Similarly, the constraint on the control variables may be expressed mathematically by

$$\gamma[\underline{u}(t)] \leq 0 \tag{1.4}$$

The block diagram of the regulator problem considered in this dissertation is shown in Figure 2. The input $\underline{r}(t)$ is identically equal to the null vector in the regulator problem and is omitted in the diagram. The output of the sampler and zero-order hold is the control

signal applied to the plant. The components of the control vector $\underline{u}(k)$ are changed at the sampling instants only. The ith component of the control vector $\underline{u}(k)$ is shown in Figure 3. Since the controls of the plant are applied in discrete steps, the plant may be described by n first order difference equations expressed mathematically by

$$\underline{x}(k) = \underline{F}[\underline{x}(k-1), \underline{u}(k), k] \tag{1.5}$$

where $\underline{F}$ is an nth order vector function. Equation (1.5) describes the states of the system at the sampling instants only. The sampling rate of the sampler must be greater than the Nyquist rate of the system to insure that the trajectory behaves properly between sampling instants (i.e., to guarantee that there are no hidden oscillations in the system).

The cost functional is evaluated at the discrete sampling instants only. Therefore, the cost integral reduces to the summation

$$J = \sum_{k=k_0}^{k_T} g[\underline{x}(k), \underline{u}(k), k] \tag{1.6}$$

where k_0 is the first sampling instant after $t=t_0$ and k_T is the last sampling instant before $t=T$.

The states of the system are constrained to lie inside the boundary at the sampling instants. This constraint can be expressed mathematically by

$$\Gamma[\underline{x}(k)] \leq 0 \tag{1.7}$$

The constraint on the control variables can be expressed mathematically by

$$\gamma[\underline{u}(k)] \leq 0 \tag{1.8}$$

In compact form, the problem considered is that of developing a method of controlling an nth order discrete-input system which is described by the vector difference equation

$$\underline{x}(k) = \underline{F}[\underline{x}(k-1), \underline{u}(k), k] \tag{1.9}$$

such that the cost functional

$$J = \sum_{k=k_0}^{k_T} g[\underline{x}(k), \underline{u}(k), k]$$

is minimized subject to the additional constraint

$$\Gamma[\underline{x}(k)] \leq 0 \tag{1.11}$$

on the system states and the additional constraint

$$\gamma[\underline{u}(k)] \leq 0 \tag{1.12}$$

on the control variables. The resulting control sequence $\underline{u}(k)$ is called the optimal control sequence or the optimal input.

In summary, there exist a number of methods for handling unbounded problems for both the continuous and discrete-input cases. Some results have been obtained for the bounded continuous problem but almost all methods involve the solution of a two-point boundary value problem. Very few methods have been given for handling the bounded state discrete-input problem and no feasible method has been presented for handling the bounded control and bounded state discrete-input problem.

CHAPTER II

PROBLEM FORMULATION AND GENERATION OF SATURATION BOUNDARY

The general discrete-input optimal control problem with constraints on the control and state variables was formulated in Chapter I. In this chapter the specific problem considered in this dissertation is formulated. A control scheme for handling this problem and its limitations are presented. The possibility of additional boundaries existing in the state space when the control variables have magnitude constraints is discussed and methods for generating these boundaries are presented. Several examples are given to illustrate the methods.

Formulation of Specific Problem

The specific problem considered in this dissertation may now be formulated as follows: compute the control variables $\underline{u}(k)$ to be applied to an nth order linear plant that can be described by the linear vector difference equation

$$\underline{x}(k) = F\underline{x}(k-1) + G\underline{u}(k) \tag{2.1}$$

such that the quadratic cost functional

$$J = 1/2 \sum_{k=k_o}^{k_o+R} [\underline{x}(k)^T P\underline{x}(k) + \underline{u}(k)^T Q\underline{u}(k)] \tag{2.2}$$

is minimized subject to the additional constraint

$$|u_i| \leq M_i, \qquad i = 1,2,\cdots,m \tag{2.3}$$

on the control variables and the constraint

$$\Gamma[\underline{x}(k)] \leq 0 \tag{2.4}$$

on the state variables. In the above formulation, Γ is a convex function in $\underline{x}$.

The constraints on the states of the system may be expressed mathematically by $\Gamma[\underline{x}] \leq 0$ and may be physical or imposed bounds as explained in Chapter I. Several different types of state constraints are shown in Figure 4. Figure 4(a) shows a magnitude bound on the x_1 state of the system. An example is an anti-aircraft gun mount on a ship that can rotate through a maximum number of degrees. The constraint on one of the states may depend on the other states. This dependency may be linear as shown in Figure 4(b) or it may be nonlinear as shown in Figures 4(c) and 4(d). As an example, consider a ground-to-air missile that has a maximum attainable velocity that increases because of fuel consumption and decreasing aerodynamic pressure as its altitude increases. Figures 4(a), (b), and (c) are examples of convex constraints on the states while Figure 4(d) is an example of a concave constraint. The convex constraint is the most common and is the one considered in this dissertation.

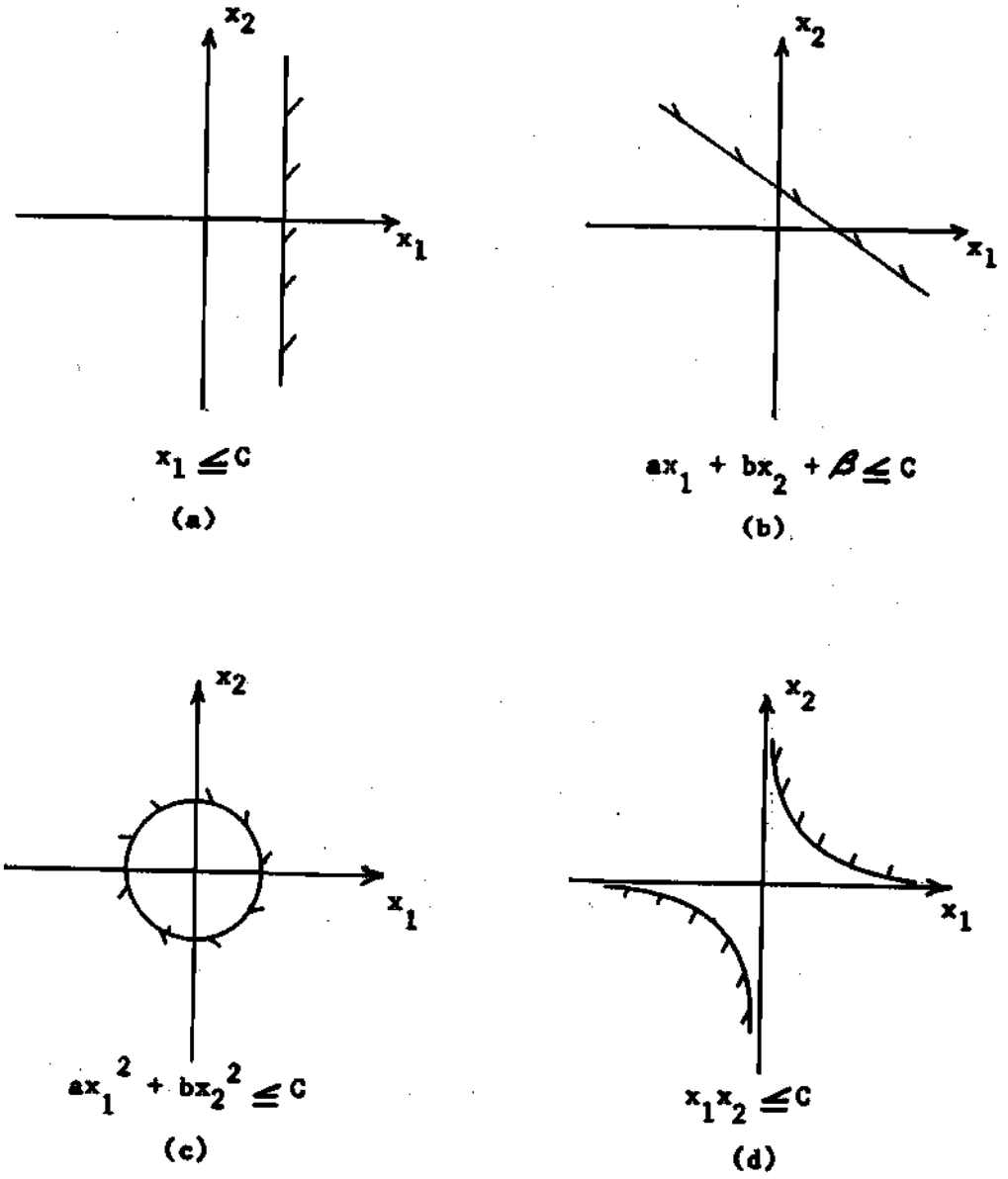


Figure 4. Types of State Variable Constraints
(a) Magnitude, (b) Linear, (c) Energy, (d) Hyperbolic.

Constraints on the control variables fall into basically two types: magnitude bounds and energy bounds. The magnitude bound on the ith component of the control vector may be expressed mathematically by

$$|u_i| \leq M_i \tag{2.5}$$

The magnitude constraint on the control variables can be represented graphically by placing a limiter in the block diagram of the system as shown in Figure 5. An energy constraint is a bound on the total energy available to control the system from time t_0 to T and may be expressed mathematically by

$$\sum_{k=k_o}^{k_T} \underline{u}(k)^T Q \underline{u}(k) \leq \text{Constant} \tag{2.6}$$

for the discrete control problem. The magnitude constraint of Equation (2.5) is the one considered in this dissertation.

The general form of the cost functional for the discrete-input problem is given by Equation (1.10). The specific cost functional considered in this dissertation is

$$J = 1/2 \sum_{k=k_o}^{k_T} [\underline{x}(k)^T P \underline{x}(k) + \underline{u}(k)^T Q \underline{u}(k)] \tag{2.7}$$

where P is an nxn positive definite matrix and Q is an mxm positive definite matrix. The quadratic form of J and the positive definiteness of the P and Q matrices insure a global minimum and therefore a unique solution. The P and Q matrices could in general depend on the value of

k. This formulation would result, for instance, if more cost were placed on the mean-square error for large values of k than for small values of k.

The cost functional in Equation (2.7) is summed from $k=k_0$ to $k=k_T$ where k_T is determined by some prespecified criterion as follows: k_T may be the first sampling instant at which the error is less than some prespecified value; k_T may be the first sampling instant at which the control energy exceeds a given value; or k_T may be a fixed integer R greater than k_0. The latter is referred to as an R-step optimal control scheme. The R-step control scheme, first proposed by Kishi (16) as a means of obtaining a practical solution to the adaptive control problem, minimizes a cost functional given by

$$J = 1/2 \sum_{k=k_o}^{k_o+R} [\underline{x}(k)^T P \underline{x}(k) + \underline{u}(k)^T Q \underline{u}(k)] \qquad (2.8)$$

The controls that minimize the cost functional over R steps are computed and the control $\underline{u}(1)$ is applied to the plant. The state of the system at the next sampling instant is computed using a model for the plant. The control to be applied during the next sampling interval is then computed by minimizing the cost functional over R steps again. This process is repeated until some criterion for completion of the problem has been satisfied. Figure 6 shows graphically how the cost for a four-step optimal control scheme is evaluated for a second order plant with one control variable.

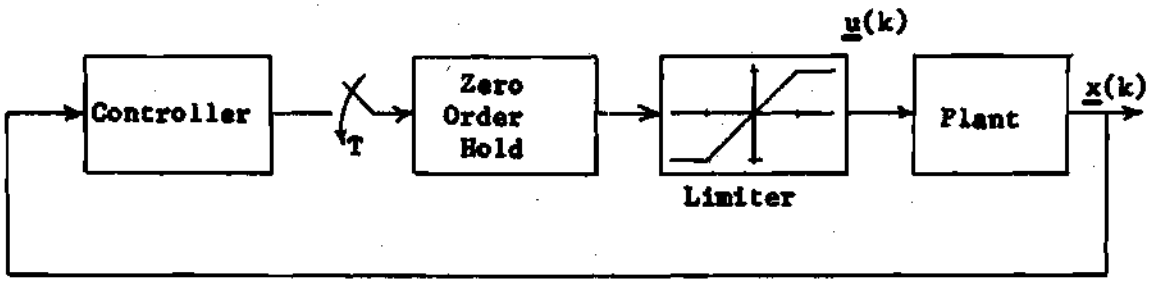


Figure 5. Discrete System with Magnitude Bound on Control Variables.

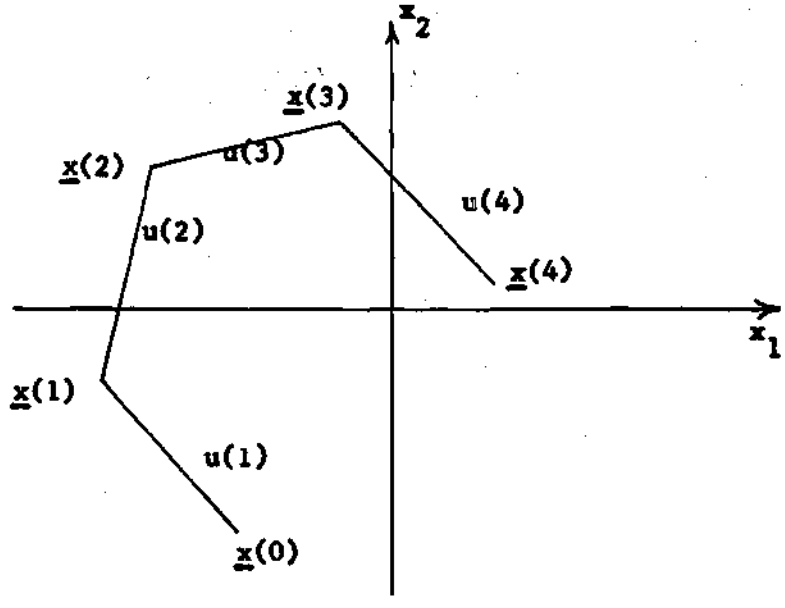


Figure 6. R-Step Optimal Controller (R = 4).

The R-step control scheme is optimal with respect to the specified cost functional in Equation (2.8) but is suboptimal with respect to the cost functional of Equation (2.7) with $k_T > k_o + R$. Therefore, if the control problem can be formulated over a fixed number of steps into the future only, such as the adaptive problem in which the system parameters change, the R-step control scheme yields an optimal solution. However, a suboptimal solution results if the R-step control scheme is used in order to obtain an on-line solution to a problem that can be formulated over all future time. This suboptimal solution can be forced to approximate the optimal solution with the proper choice of R and sampling period as explained in Chapter V.

The problem may be formulated as a tracking or as a regulator problem. In this dissertation, only the regulator problem is considered. Tracking problems in which the reference input is completely described can be reformulated into an equivalent regulator problem with a few modifications on the controller.

Saturation Boundary

In the problem with bounds on the state variables but with no bounds on the control variables, there is sufficient control available to prevent the system trajectory from crossing the state boundary from any initial point that lies in the allowable region. With magnitude bounds on the control variables, however, there may exist regions in the state space, which lie below the original boundary, from which a trajectory cannot be prevented from crossing the original state boundary with the control available. Any initial condition on $\underline{x}$ which lies in

the original unallowable region or in one of these additional unallowable regions will cause the problem to have an unacceptable solution. In an infinite-step optimal controller (i.e., one that computes the complete optimal sequence $\underline{u}(k)$ at one computation), the trajectory will never enter one of these additional unallowable regions. In the R-step optimal controller, however, the trajectory might be permitted to enter one of the additional unallowable regions if the Rth future point does not cross the original state boundary (i.e., the controller does not "see ahead" far enough to know that the boundary is being approached). Since this results in an unacceptable solution, it follows that the additional unallowable regions must be computed and considered in the design of the R-step optimal controller.

Generation of Saturation Boundary

The general technique for finding the additional unallowable regions in the state space that exist when the system has bounds on the state variables and magnitude bounds on the control variables is presented in this section. The optimal method for considering these additional regions in the design of an R-step optimal controller is presented in Chapter III.

The additional unallowable regions can be described by their boundary with the allowable regions. This boundary is the locus of all points from which the original boundary can just be avoided using the maximum allowable control. Because of the manner in which it is defined, this boundary is referred to as the saturation boundary (S.B.). For the linear system considered in this dissertation, the general mathematical

expression for the S.B. is

$$\underline{x} = \phi(\underline{x}_0, \underline{u}, t) \tag{2.9}$$

where ϕ is differentiable with respect to each of the independent variables. Therefore, the S.B. is a smooth surface which is tangent to the original boundary $\Gamma[\underline{x}]=0$. Since the original boundary is convex in $\underline{x}$, the S.B. is also convex in $\underline{x}$ with respect to the allowable region.

The first step in determining the S.B. is to find an expression for the tangency points with the original boundary. This expression must be obtained from the continuous equations of the plant. Since the gradient $\nabla\Gamma[\underline{x}]$ of the original boundary is everywhere perpendicular to the boundary and since the velocity $\dot{\underline{x}}$ of the trajectory using maximum control is everywhere tangent to the trajectory, the solution of

$$\nabla\Gamma[\underline{x}] \cdot \dot{\underline{x}} = 0 \tag{2.10}$$

yields all points on all possible maximum control trajectories at which the trajectory is parallel with the original boundary. Since all points of tangency lie on the original boundary, they must also satisfy the equation

$$\Gamma[\underline{x}] = 0 \tag{2.11}$$

The solution of Equations (2.10) and (2.11) simultaneously for $\underline{x}$ yields an analytical expression for the tangency points.

The tangency points may also be found by using the discrete equations of the plant given by

$$\underline{x}[(n+1)T] = F\underline{x}[nT] + G\underline{u}[(n+1)T] \tag{2.12}$$

The point of tangency is defined to occur when two successive points on a trajectory using maximum control lie on the original boundary. Stated mathematically

$$\Gamma[\underline{x}[nT]] = 0 \tag{2.13}$$

and

$$\Gamma[\underline{x}[(n+1)T]] = 0 \tag{2.14}$$

These two points may be computed by substituting $\underline{x}[(n+1)T]$ from Equation (2.12) into Equation (2.14) and then solving the resulting equation simultaneously with Equation (2.13). Points on the saturation boundary may then be found by solving Equation (2.12) backwards in time from the points of tangency. This is done by successively solving for $\underline{x}[nT]$ from the equation

$$\underline{x}[nT] = F^{-1}\underline{x}[(n+1)T] - F^{-1}G\underline{u}[(n+1)T] \tag{2.15}$$

The difference between the points of tangency computed using the continuous equations and using the discrete equations of a second order plant is illustrated in Figure 7. Point A is the point of tangency

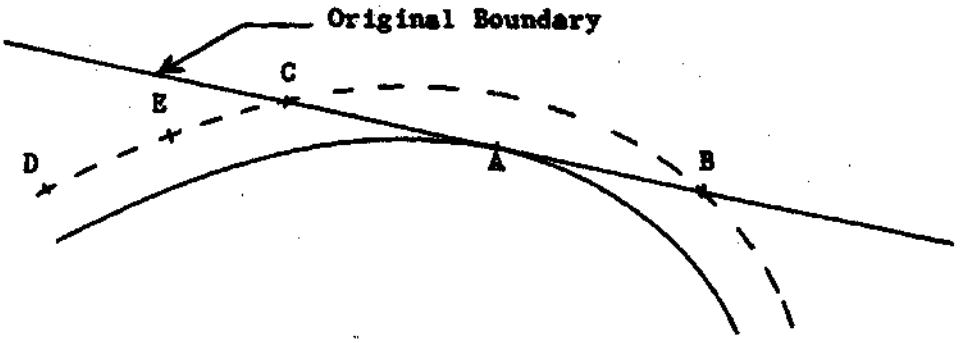


Figure 7. Comparison of Generation of Tangency Points from Discrete and Continuous Equations.

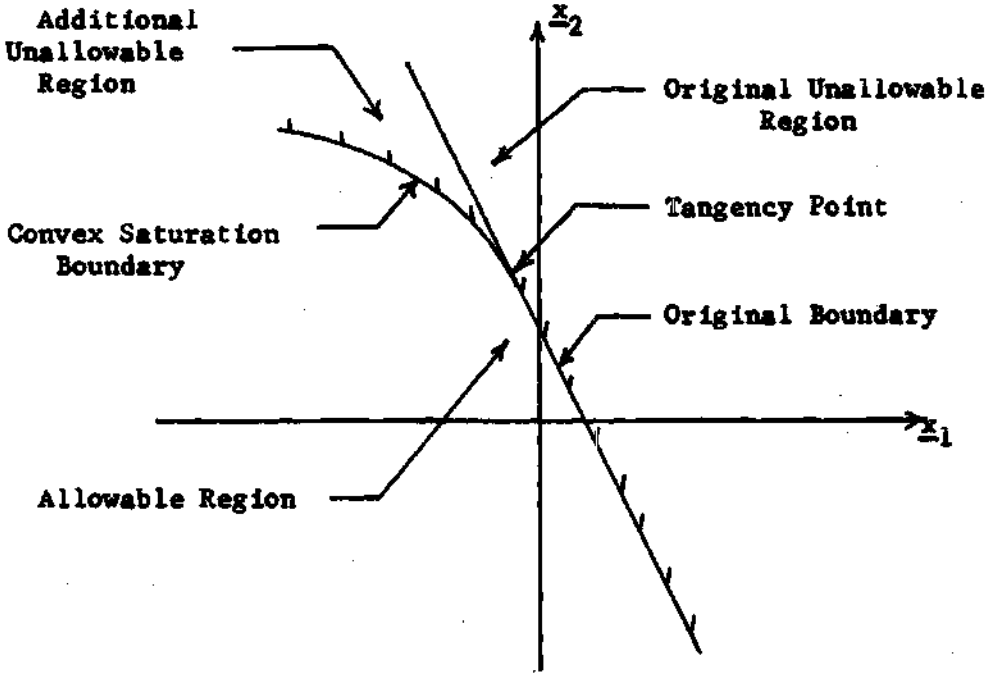


Figure 8. Saturation Boundary for a Second Order System.

computed using the continuous equations. Points B and C are the points computed using the discrete equations. The point A will always fall between points B and C with the spread between B and C becoming smaller as the sampling period used in generating the discrete equations is decreased. In the limit as $T \to 0$ the points A, B, and C will coincide. Point D is a point on the S.B. one sampling period from point C and is found by solving Equation (2.15).

Note that the trajectory using maximum control exceeds the original boundary between the points C and B in Figure 7. This is caused by the fact that when using the discrete equations, the points on the trajectory at the sampling instants only are constrained to lie in the allowable region. Suppose that the S.B. generated from the discrete equations is used and a sampling instant on an optimal trajectory occurs at point E. It is obvious that because of the magnitude constraint on the control variables, the trajectory at the next sampling instant will not be in the allowable region. It follows that the S.B. must be generated using the points of tangency found from the continuous equations of the plant.

For an nth order system, the S.B. is an n-dimensional hypersurface in the state space. The points of tangency of the S.B. with the original boundary form an (n-1)-dimensional hypersurface in state space. For example, consider a second order plant with a linear state constraint on $\underline{x}$ as shown in Figure 8 (p. 20). The S.B. is a two-dimensional surface, a curve, and the tangency of the S.B. with the original linear state boundary is a one-dimensional surface, a point. This concept may be extended to a third order plant where the S.B. is a three-dimensional

surface and the points of tangency form a two-dimensional curve. For systems of higher order than three, the surface becomes an n-dimensional hypersurface.

An analytical expression for the S.B. is in general difficult to find. As an example, consider a time-invariant linear plant described by

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \tag{2.16}$$

where A is an nxn matrix and B is an nxm matrix. Taking the Laplace transform of the above equation yields

$$s\underline{X}(s) - \underline{x}_0 = A\underline{X}(s) + B\underline{U}(s)$$

or

$$\underline{X}(s) = (sI-A)^{-1}B\underline{U}(s) + (sI-A)^{-1}\underline{x}_0 \tag{2.17}$$

where $\underline{x}_0$ is the initial point in state space and $\underline{U}(s)$ is the Laplace transform of the maximum controls which may be given by $\pm U_{max}/s$. The inverse Laplace transform of Equation (2.17) gives the equation of each of the state variables as a function of time t and the initial state $\underline{x}_0$ as follows:

$$\underline{x}(t) = \mathcal{L}^{-1}[(sI-A)^{-1}B\underline{U}(s) + (sI-A)^{-1}\underline{x}_0] \tag{2.18}$$

In Equation (2.18), if $\underline{x}(t)$ is forced to satisfy the analytical

expression for the tangency points, then each of the $\underline{x}_0$ components may be expressed as a function of t and the system characteristic roots. For many systems these n equations may be expressed as one equation relating the initial points that have trajectories that are tangent to the original boundary. This equation is the analytical expression for the saturation boundary. Since the n equations for the initial conditions $\underline{x}_0$ are in general transcendental, one equation relating them is difficult to find. An alternate method of representing the S.B. is presented in the next section of this chapter.

Approximation of Saturation Boundary

As pointed out in the previous section, an analytical expression for the S.B. is not always possible to find because of the transcendental nature of the state equations. An alternate method of representing the S.B. is to compute points on the S.B. and use an approximation to the boundary between the points. The simplest and most convenient approximation for the S.B. of an nth order system is nth order hyperplanes connecting all adjacent n points. Although this seems to be a coarse approximation, it may be made as accurate as desired by taking the points on the S.B. closer together. Eventually there has to be a trade-off between the accuracy desired and the additional storage space required in the controller.

One method for computing the points on the S.B. is to integrate the difference equations backward in time from the points of tangency using Equation (2.15). The sampling period used to generate the S.B. should be small compared with the sampling period of the system to

obtain reasonable accuracy. A second order example for a system with complex roots is shown in Figure 9.

Examples

Consider a plant with a transfer function given by

$$G(s) = 1/s(s+1) \tag{2.19}$$

The system is second order with real characteristic roots. The state variable formulation of the problem is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.20}$$

For a sampling period of T=1 second, the difference equations are

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} 1 & .632 \\ 0 & .368 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} .368 \\ .632 \end{bmatrix} u(k) \tag{2.21}$$

The constraint on the control variable is given by

$$|u| \leq U_{max} \tag{2.22}$$

where U_{max}=1 and the linear constraint on the state variables is given by

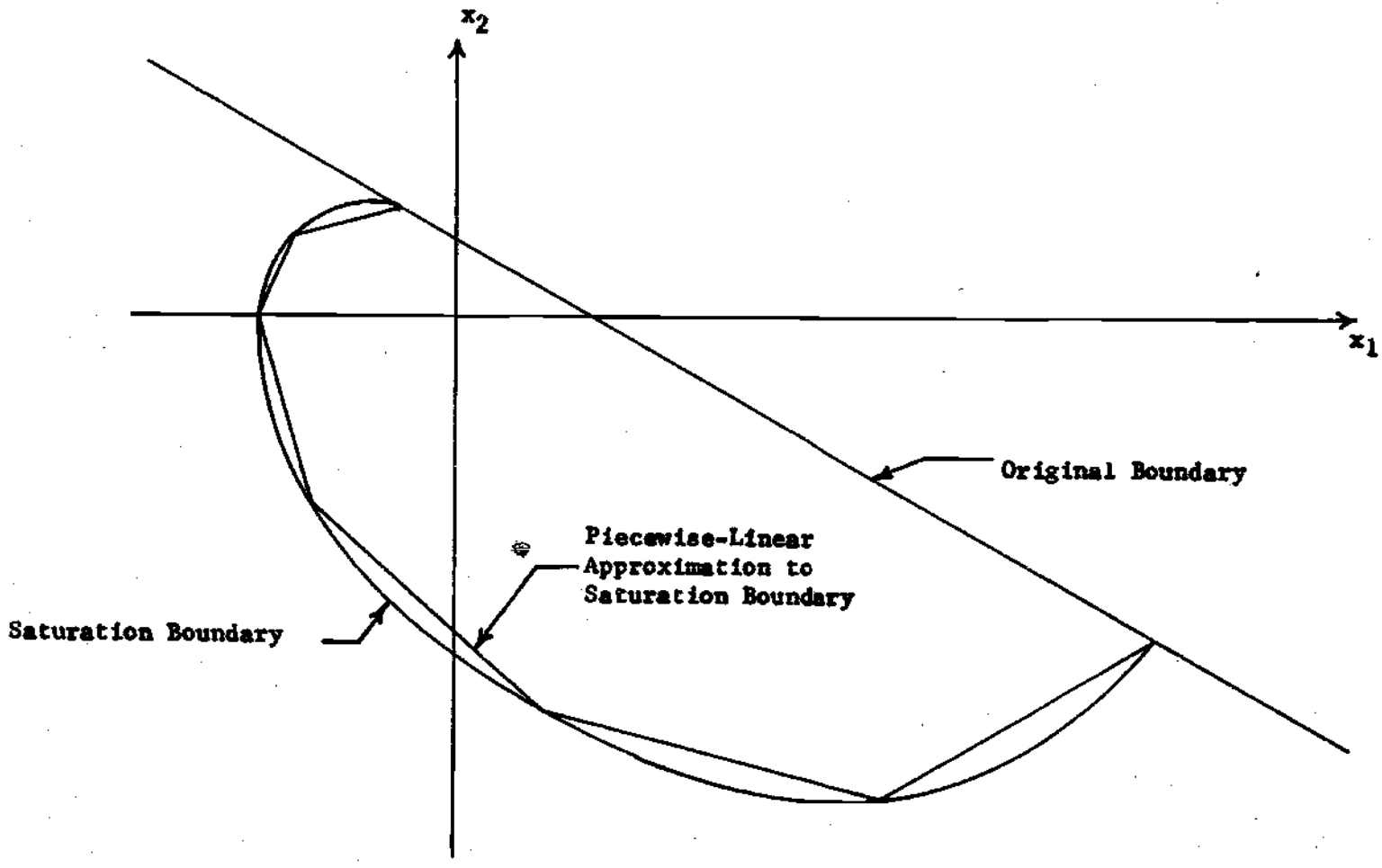


Figure 9. Saturation Boundary with Approximation.

$$\underline{\alpha}^T \underline{x}(k) + \beta \leq 0 \tag{2.23}$$

where $\underline{\alpha}^T = [2 \quad 1]$ and $\beta = -1$.

Solving Equations (2.10) and (2.11) simultaneous yields $x_1=0$, $x_2=1$ as the point of tangency of the S.B. with the original linear boundary. Using Equation (2.18) with $u = -U_{max} = -1$, the equations for the states of the system in the time domain are

$$\begin{aligned} x_1(t) &= 1 - t - \varepsilon^{-t} + x_{10} + (1-\varepsilon^{-t})x_{20} \\ x_2(t) &= -1 + \varepsilon^{-t} + (\varepsilon^{-t})x_{20} \end{aligned} \tag{2.24}$$

The u=+1 trajectory that is tangent to the original boundary is above the boundary and therefore need not be considered. An analytical expression for the S.B. may be obtained by substituting the tangency point for $x_1(t)$ and $x_2(t)$ in Equations (2.24) and solving the two resulting equations simultaneously for x_{10} in terms of x_{20} with the time variable eliminated.

$$x_{10} = 1 - x_{20} + \ln[1/2(1+x_{20})] \tag{2.25}$$

The results are shown in Figure 10. The same system with a parabolic constraint on the states is shown in Figure 11.

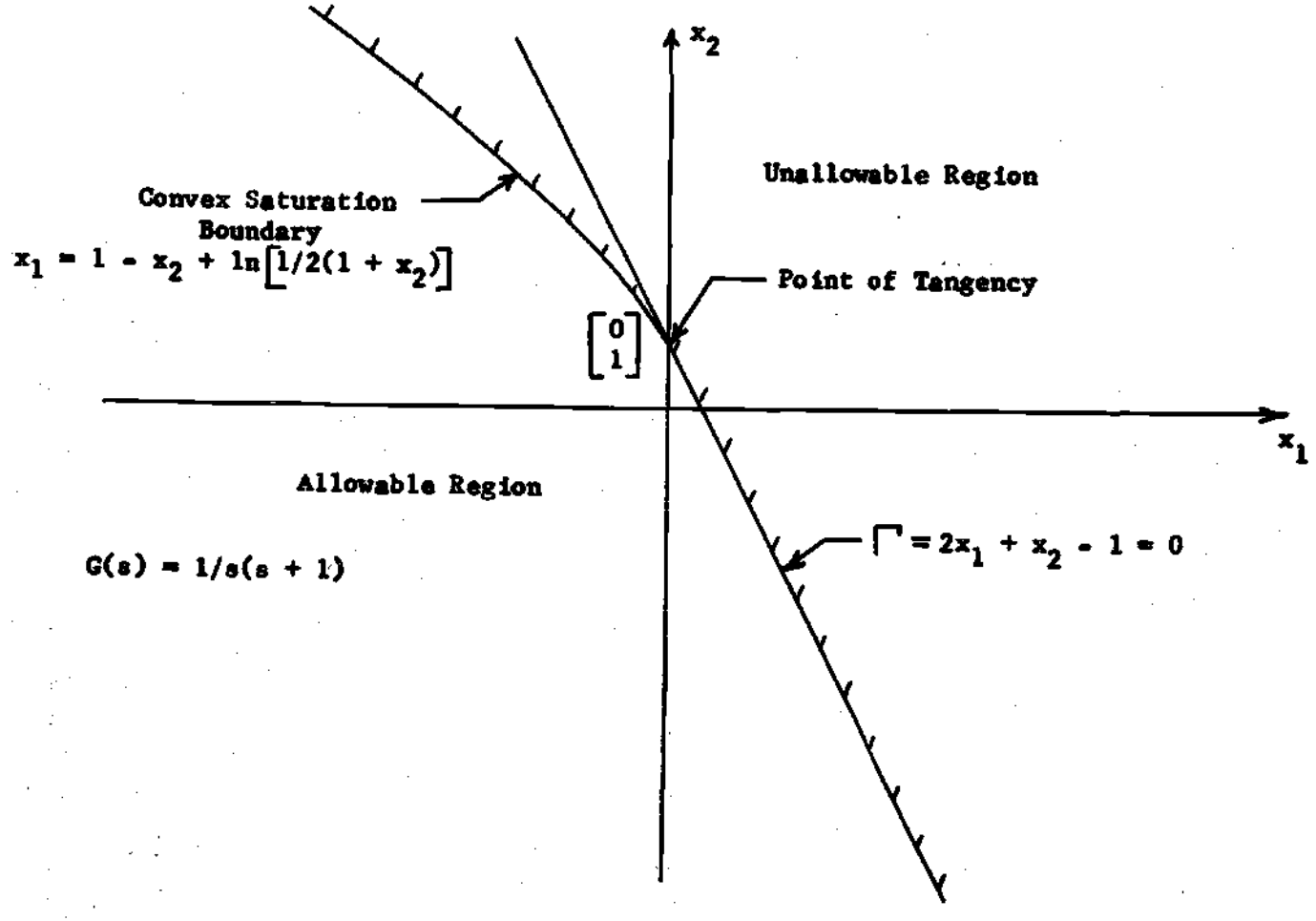


Figure 10. Saturation Boundary for System with Real Roots and a Linear Constraint on the States.

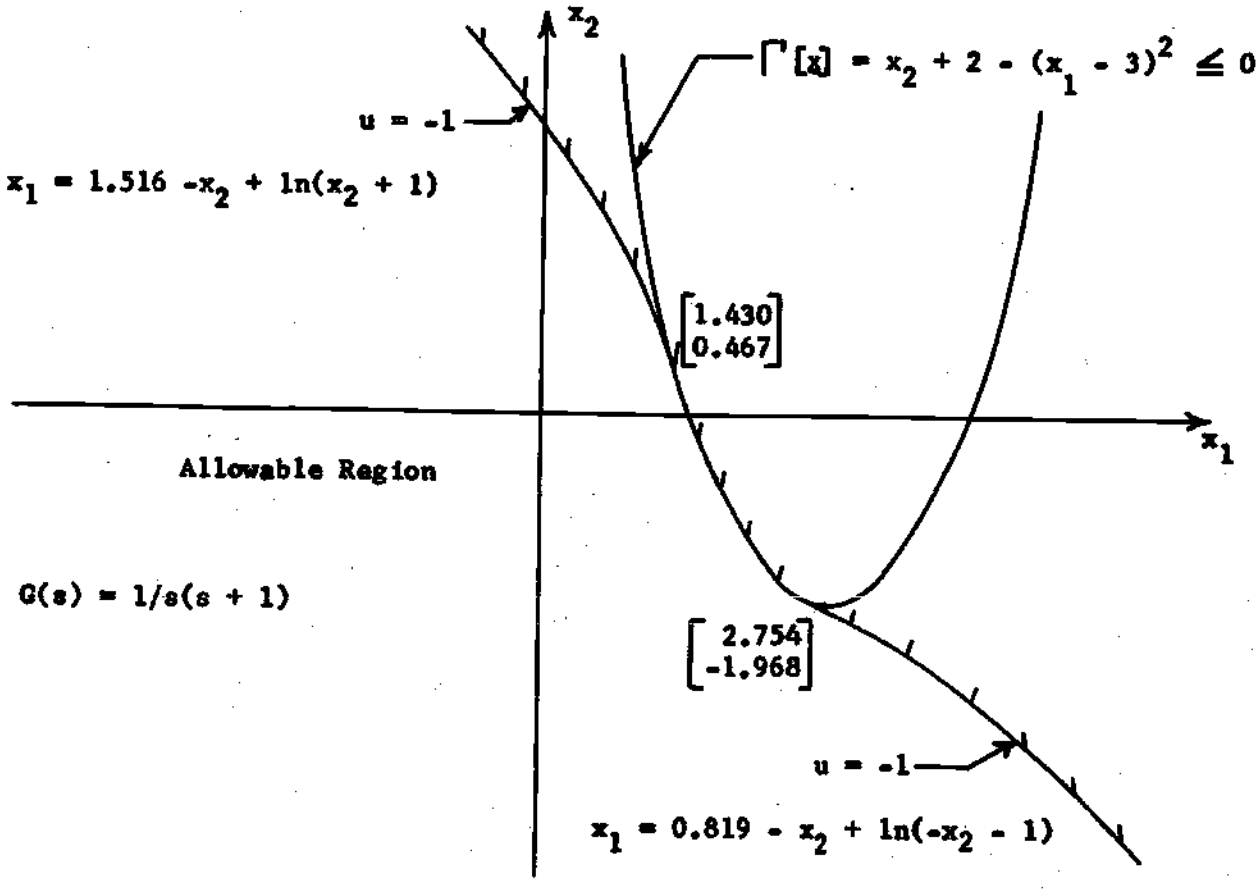


Figure 11. Saturation Boundary for System with Real Roots and a Parabolic Constraint on the States.

As expected, systems with complex roots have S.B. that spiral outward from the point of tangency. Figure 12 shows such a system with a linear constraint on the states. No analytical expression for the S.B. could be obtained for this case because of the transcendental nature of the state equations. The same system with a hyperbolic constraint on the states is shown in Figure 13.

In summary, the specific problem considered in this dissertation was formulated and the types of constraints considered were discussed in this chapter. The R-step optimal controller and its limitations were discussed. The reasons for needing additional boundaries in the state space and methods for generating these boundaries were presented. A technique for considering the boundaries in state space and the basic theory of the optimization algorithm will be presented in Chapter III.

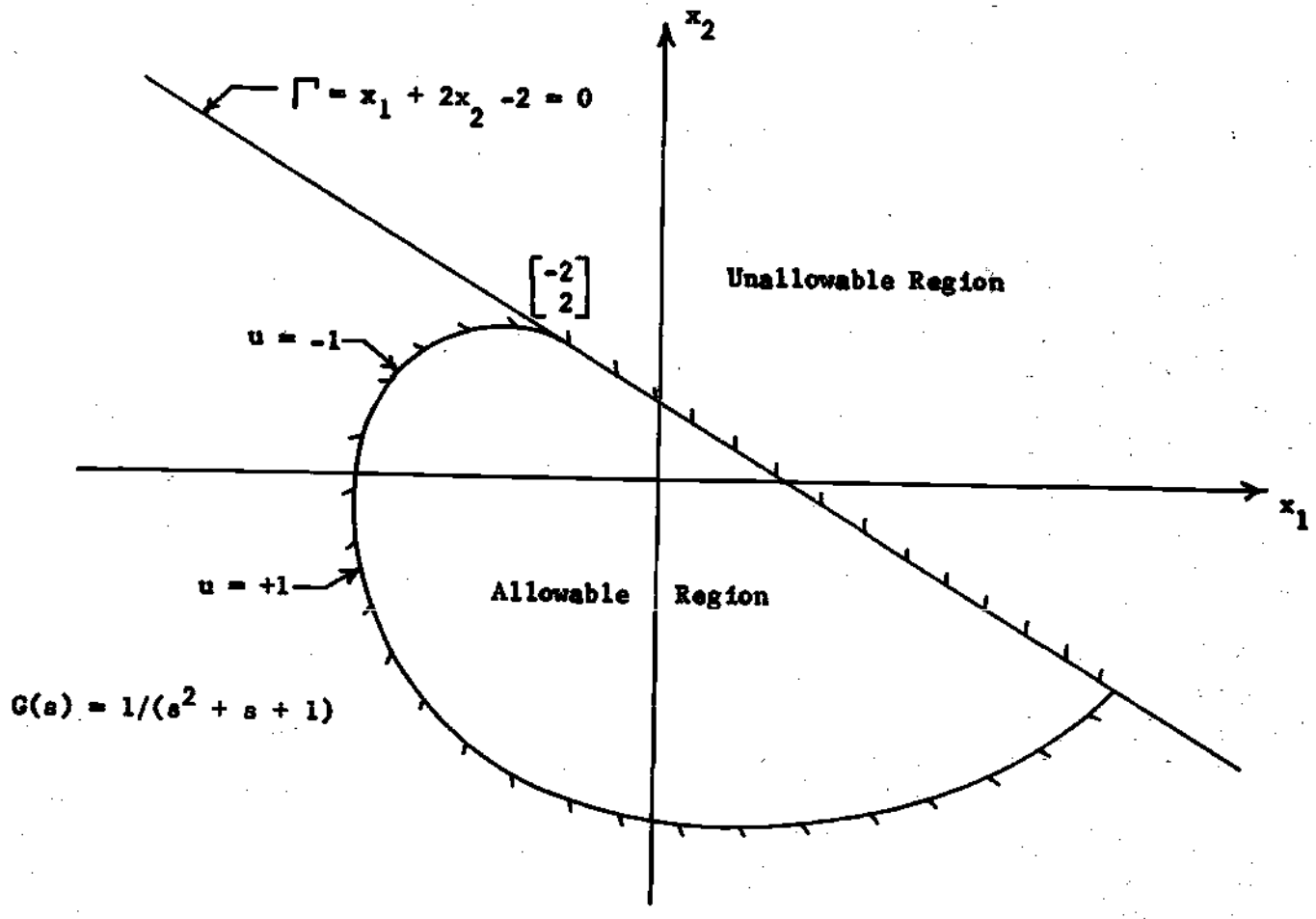


Figure 12. Saturation Boundary for System with Complex Roots and a Linear Constraint on the States.

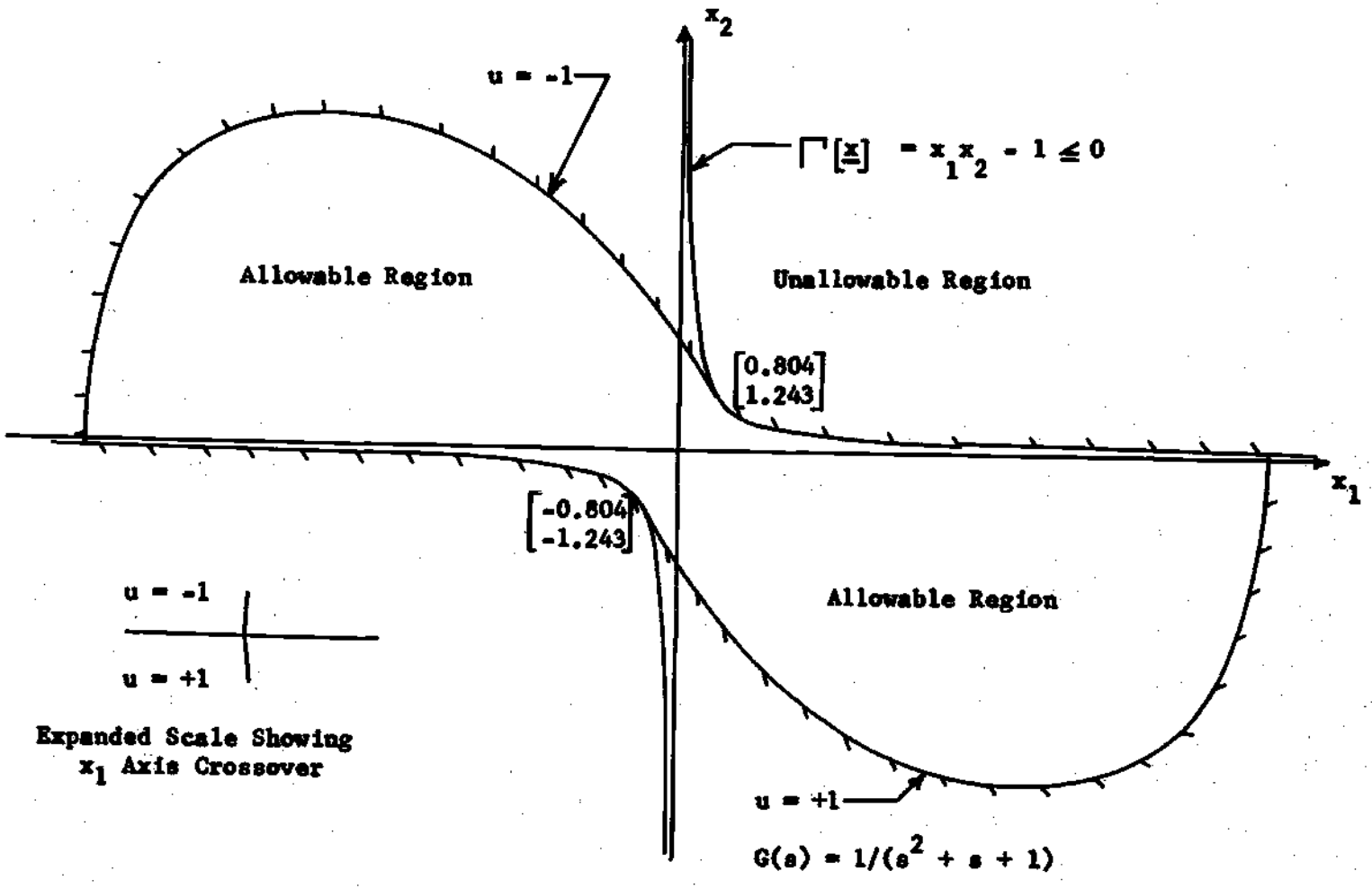


Figure 13. Saturation Boundary for System with Complex Roots and a Hyperbolic Constraint on the States.

CHAPTER III

BASIC THEORY AND OPTIMIZATION ALGORITHM

As pointed out in Chapter II, when using an R-step optimal controller, the trajectory must not be allowed to violate the original boundary or the saturation boundary. A theorem on optimality for systems with an R-step controller and convex constraints on the states is presented and proved in this chapter. An optimization algorithm for the above system is also presented in this chapter.

Theorem on Optimality

An R-step optimal controller computes the set of controls $\underline{u}(1)$ through $\underline{u}(R)$ which minimizes the cost functional of Equation (2.2) while satisfying the control constraints of Equation (2.3) and the state constraints of Equation (2.4). It was shown in Chapter II that if the original convex boundary of Equation (2.4) is not to be violated, the convex S.B. must also not be violated. Suppose the set of control variables which minimizes the cost functional forces one or more points on the trajectory to lie above the convex state boundary in one of the unallowable regions as shown in Figure 14. Since this is an unacceptable solution, the

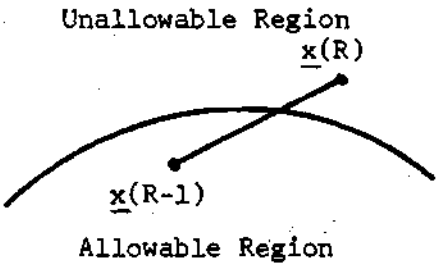


Figure 14. Example of Trajectory which Violates the State Constraint

controller must recompute a set of controls which forces all R points on the trajectory to lie on or below the state boundary. Out of the set of all such possible solutions, the controller must compute the one that minimizes the cost functional of Equation (2.2).

Theorem I: Convexity of Constant Cost Contours in State Space

The constant cost contours in state space for linear systems with quadratic cost indices and magnitude bounds on the control variables are convex.

For proof of Theorem I, consider the system in Figure 15 in which the point marked $\underline{x}(R)$ is the end point of the trajectory that has minimum cost $J=c$ over R steps. Consider a hyperplane in this state space described by

$$\underline{\alpha}^T\underline{x} + \beta = 0 \qquad (3.1)$$

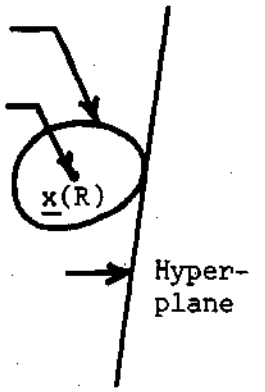


Figure 15. Constant Cost Contour with Tangent Hyperplane

To force the point $\underline{x}(R)$ to lie on this hyperplane, $\underline{u}(R)$ is constrained by solving

$$\underline{\alpha}^T\underline{x}(R) + \beta = 0 \qquad (3.2)$$

$$\underline{x}(R) = F\underline{x}(R-1) + G\underline{u}(R)$$

for $u(R)_p$ and substituting the resulting expression into the cost expression, thereby eliminating $u(R)_p$. The resulting cost expression is

quadratic in $\underline{u}(1)$ through $\underline{u}(R)'$ and $\underline{x}(1)$ through $\underline{x}(R)$ and therefore has a unique minimum. The locus of all possible trajectories that have this same cost, $J=c+\Delta c$, is a closed smooth surface in this state space as shown in Figure 15 (p. 33). Because of the quadratic nature of the reduced cost functional, the cost is a monotonically increasing function as the end point $\underline{x}(R)$ is forced on the hyperplane away from the point yielding minimum cost. From this result, it is obvious that the hyperplane is a supporting hyperplane to the constant cost contour $J = c+\Delta c$. Since the hyperplane was chosen arbitrarily and could be any tangent hyperplane to the contour $J=c+\Delta c$, it follows that the constant cost contours in the state space are convex.

Theorem II: Optimal Method for Constraining Trajectory for Systems with Convex Constraints on the States

Consider the linear system with constraints on the control and state variables as described by Equations (2.1) through (2.4). Points on an unbounded trajectory that fall in an unallowable region should be bounded to the convex state boundary. Proof that this procedure yields an optimal solution (i.e., minimizes the total cost) is given below.

Discussion of Theorem II. Assume that a method is available for bounding a point on the trajectory to a convex boundary. In Figure 16 let $\hat{u}$ be the set of controls that minimizes the quadratic cost functional satisfying the magnitude constraints on the controls. Henceforth, a set of controls that satisfies the magnitude constraint on the control variables will be referred to as an admissible set of controls. The point at which $\hat{u}$ is shown represents the Rth point on the trajectory computed by the R-step optimal controller and lies in an unallowable region.

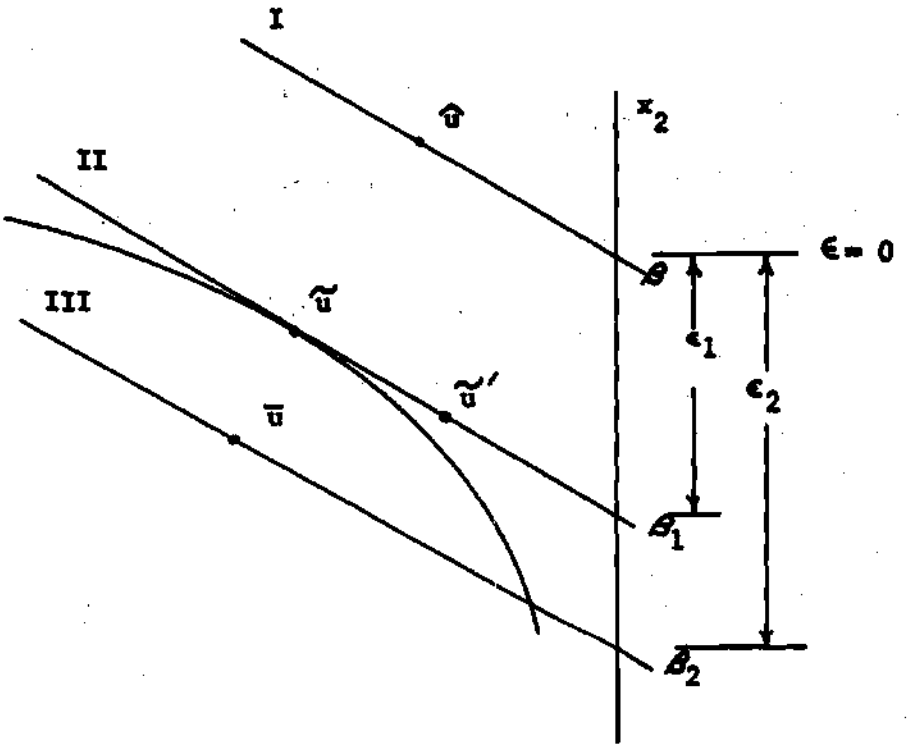


Figure 16. Possible Sets of Admissible Minimizing Controls.

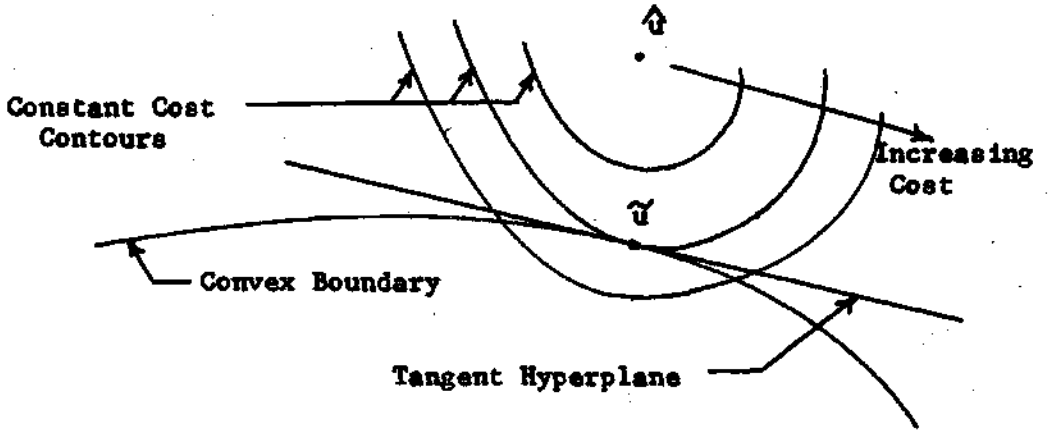


Figure 17. Minimizing Control Sequence $\tilde{u}$ with States Bounded.

The Rth point was chosen as the unacceptable point for convenience in the proof but could be any of the R points on the trajectory computed by the controller. Since the solution is not acceptable if any point on the trajectory falls in an unallowable region, the Rth point must be bounded to the boundary or to some point below the boundary. Let $\tilde{u}$ be the set of admissible controls that minimizes the cost functional with the Rth point constrained to lie on the convex boundary. Let $\bar{u}$ be the set of admissible controls that minimizes the cost functional with the Rth point constrained to lie below the convex boundary in the allowable region. To prove the theorem, it is sufficient to prove that

$$J[\hat{u}] < J[\tilde{u}] < J[\bar{u}] \tag{3.3}$$

Since the set $\hat{u}$ was computed without any constraints on the states of the system, it follows that

$$J[\hat{u}] < J[\tilde{u}]$$

and (3.4)

$$J[\hat{u}] < J[\bar{u}]$$

Therefore, to prove the theorem, it must be shown that

$$J[\tilde{u}] < J[\bar{u}] \tag{3.5}$$

In Figure 16 (p. 35) a tangent hyperplane to the convex boundary is constructed at the point $\tilde{u}$. The point $\tilde{u}$ is the end point of the trajectory that has the smallest cost of all trajectories that have their end points constrained to lie on the convex boundary. Because of the convexity of the state boundary and the convexity of the constant cost contours, $\tilde{u}$ is the point of tangency of the state boundary with the constant cost contour $J[\tilde{u}]$. Since the constructed hyperplane is tangent to the state boundary at $\tilde{u}$, it is also tangent to this constant cost contour. But from the results of Theorem I, if the end point of the trajectory is constrained to lie on the tangent hyperplane, the cost monotonically increases as the end point is forced away from this point of tangency. Therefore, the set of controls $\tilde{u}$ also yields minimum cost if the end point is constrained to the tangent hyperplane. This result is illustrated for a second order system in Figure 17 (p. 35).

Additional hyperplanes parallel to the above hyperplane are constructed through the end points corresponding to the set of controls $\hat{u}$ and $\bar{u}$. Since $\hat{u}$ is the set of controls yielding minimum cost without any constraints on the states, it follows that $\hat{u}$ is the set of admissible controls giving minimum cost if the end point is constrained to lie on the hyperplane through the point $\hat{u}$. Since $\bar{u}$ was assumed to be the set of controls yielding minimum cost if the end point is constrained to lie below the convex boundary, it follows that the set of controls $\bar{u}$ minimizes the cost if the end point is constrained to lie on the hyperplane through the point $\bar{u}$.

Proof of Theorem II. The three hyperplanes in Figure 17 (p. 35) may be described by

$$\begin{aligned} \text{I:} \quad & \underline{\alpha}^T \underline{x} + \beta = 0 \\ \text{II:} \quad & \underline{\alpha}^T \underline{x} + \beta_1 = 0 \\ \text{III:} \quad & \underline{\alpha}^T \underline{x} + \beta_2 = 0 \end{aligned} \tag{3.6}$$

or equivalently by

$$\begin{aligned} \text{I:} \quad & \underline{\alpha}^T \underline{x} + \beta = 0 \\ \text{II:} \quad & \underline{\alpha}^T \underline{x} + \beta + \varepsilon_1 = 0 \\ \text{III:} \quad & \underline{\alpha}^T \underline{x} + \beta + \varepsilon_2 = 0 \end{aligned} \tag{3.7}$$

Note that the $\underline{\alpha}$ vector is fixed by the slope of the hyperplane and is the same in each equation. To constrain the end point of a trajectory to one of the hyperplanes, a component of $\underline{u}(R)$ may be constrained. The remaining components of $\underline{u}(R)$ are denoted by $\underline{u}(R)'$. The constrained component of $\underline{u}(R)$ is linear in β and may be obtained from Equations (3.2).

$$u(R)_p = f[\underline{x}(R-1), \underline{u}(R)', \beta] \tag{3.8}$$

Substituting this required expression for $u(R)_p$ into the quadratic cost

functional and successively eliminating $\underline{x}(R)$, $\underline{x}(R-1)$, etc. by using the difference equation, the cost turns out to be a quadratic function of β and $\underline{u}(1)$ through $\underline{u}(R)'$.

$$J = 1/2\{\beta^2 + \beta f_1[\underline{u}(1), \underline{u}(2), \cdots, \underline{u}(R)', \underline{x}(0)] \tag{3.9}$$

$$+ f_2[\underline{u}(1)^T C_1 \underline{u}(1), \cdots, \underline{u}(R)'^T C_R \underline{u}(R)', \underline{u}(1), \cdots, \underline{u}(R)', \underline{x}(0)]\}$$

Differentiating with respect to each control vector component and equating to zero yields

$$\partial J/\partial u(k)_j = a_{jk}\beta + b_{jk}u(k)_j + \sum_{\substack{i=1 \\ i\neq j}}^{R} b_{ik}u(k)_i + c_{jk} = 0 \tag{3.10}$$

for $k = 1,2,\cdots,R$; $k=R$, $j\neq p$

Solving the above equations for each control component yields

$$u(k)_j = a'_{jk}\beta + b'_{jk} \tag{3.11}$$

where a'_{jk} and b'_{jk} are constants. Each control component is a linear function of β. Substituting each of the above expressions of the control components into the cost expression, yields a cost that is quadratic in β and therefore quadratic in ε. But from Equation (3.2) the cost is a minimum for $\varepsilon=0$. Since the cost is quadratic in ε and has a minimum, it is convex as shown in Figure 18. It follows that

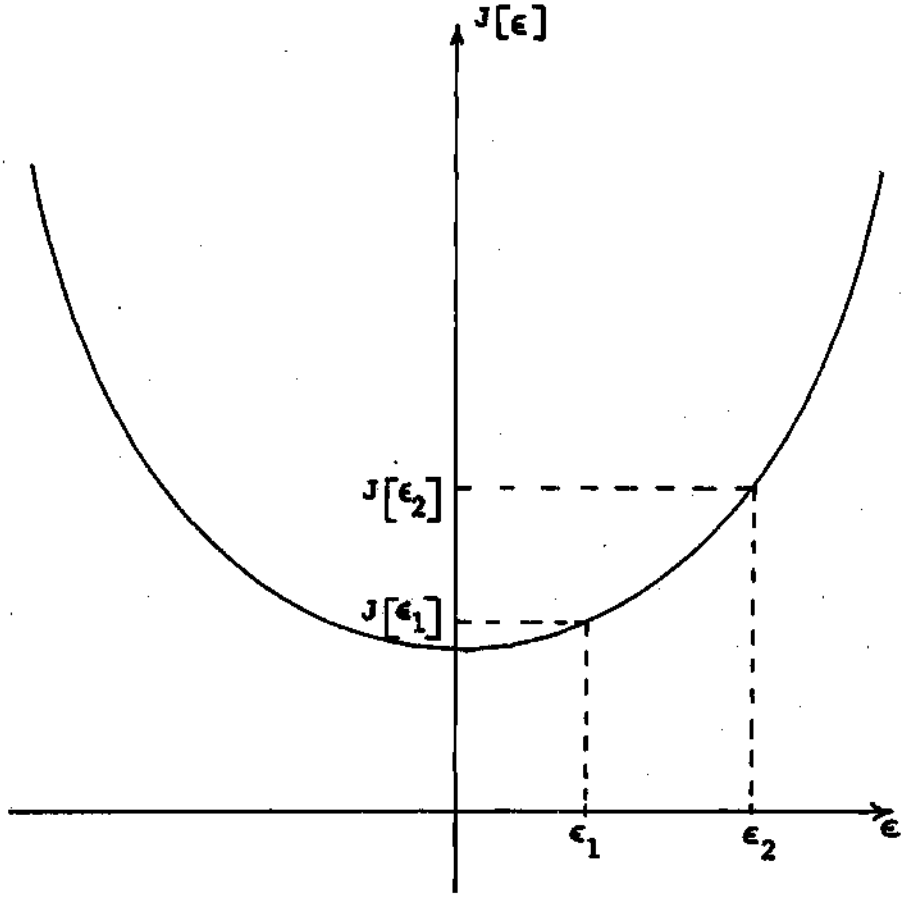


Figure 18. Quadratic Cost in ϵ.

$$J[\varepsilon_1] < J[\varepsilon_2]$$

or

$$J[\tilde{u}] < J[\bar{u}] \tag{3.12}$$

since $\varepsilon_1 < \varepsilon_2$.

In the proof of Theorem II, it was necessary that the set of controls $\hat{u}$, $\tilde{u}$, and $\bar{u}$ be the sets that give minimum cost if the end point of the trajectory is constrained to lie on the three respective hyerplanes. If these were not optimal sets of controls, then to compare the three costs on the basis of their convexity in ε would be meaningless.

In the derivation of Equation (3.11), it was assumed that none of the control variables exceeds the saturation value. Suppose that some of the optimal control variables have to be bounded. The remaining control variables are still linear functions of β and the cost is still quadratic in β and ε. If the same control variables are bounded in the set $\tilde{u}$ as in the set $\bar{u}$, then it follows that $J[\tilde{u}] < J[\bar{u}]$. Suppose that the control variables of $\tilde{u}$ and $\bar{u}$ that are bounded are not the same, then the above argument breaks down. However, if the unacceptable point is constrained to lie on the hyperplane corresponding to β_1 and the same controls are set at the saturation value as in the $\bar{u}$ set of controls, then the resulting set of optimal controls $\tilde{u}'$ from the above results are such that $J[\tilde{u}'] < J[\bar{u}]$. But $J[\tilde{u}] < J[\tilde{u}']$, since $\tilde{u}'$ was computed with one more constraint than $\tilde{u}$. Therefore, $J[\tilde{u}] < J[\bar{u}]$ for all possible combinations of bounded control.

The R-step optimal control scheme minimizes the cost functional over R steps into the future with or without the states bounded and applies the control $\underline{u}(1)$ during the next sampling interval. The procedure is then repeated to compute the control vector to be applied during the following sampling period. If points further than one step into the future are being bounded in the optimization algorithm, the question arises as to whether this procedure gives any better results than the procedure of ignoring the boundary until the point $\underline{x}(1)$ is found to violate the boundary. The trajectory resulting from this procedure, however, is in the class of possible solutions using the R-step optimal control scheme described above. Therefore, the cost of the R-step optimal control scheme is less than or equal to the cost of the scheme of ignoring the boundary until the point $\underline{x}(1)$ violates it.

Optimization Algorithm

There are three distinct possibilities that can arise in the optimal control of systems with bounded controls and bounded state variables: both the controls and states remain within the prescribed bounds; states remain within the prescribed bounds but the controls do not; states do not remain in the prescribed bounds. In the third class the control variables may or may not be bounded. Each of the three above possibilities requires a separate procedure in the optimization algorithm.

Controls and States Within Prescribed Bounds

If the control and state variables remain within their prescribed bounds, the optimization algorithm is the same as for the problem with

no bounds imposed. The quadratic cost functional of Equation (2.2) is reduced to a function of the control variables and the initial state only as shown in Equation (3.13) by the successive substitution of the difference Equation (2.1).

$$J = f[\underline{u}(1), \underline{u}(2), \cdots, \underline{u}(R), \underline{x}(0)] \tag{3.13}$$

The set of controls that minimizes Equation (3.13) can be obtained by using ordinary calculus. Differentiating Equation (3.13) with respect to each control component, equating to zero, and then solving yields

$$\begin{aligned}
&\partial J/\partial u(1)_1 = 0 \qquad && u(1)_1 = C_1 \\
&\partial J/\partial u(1)_2 = 0 \qquad && u(1)_2 = C_2 \\
&\quad \vdots \qquad && \quad \vdots \\
&\partial J/\partial u(R)_m = 0 \qquad && u(R)_m = C_{Rm}
\end{aligned} \tag{3.14}$$

where $u(j)_i$ is the ith component of the control vector during the jth sampling interval. Each control component is equal to a constant that is a function of the dynamics of the plant and the initial condition $\underline{x}_0$. For a time-invariant plant, these controls can be stored as a function of $\underline{x}_0$ only and computed very rapidly by the controller.

States Within Prescribed Bounds--Controls Not Within Prescribed Bounds

If one or more of the unconstrained optimal controls in Equation (3.14) exceeds its magnitude bound, the cost functional of Equation (3.13) must be reminimized with the magnitude constraint added. There are a number of methods of computing these constrained controls. An iterative procedure, called the coordinatewise gradient technique and used by Kishi in reference (16), is used in this dissertation because of its fast convergence and easy implementation.

Stated mathematically, the (n+1)th iteration of the jth control vector is given by

$$\underline{u}(j)^{(n+1)} = \underline{u}(j)^{(n)} + \varepsilon_{n_j} \nabla_j{}^{(n)} \tag{3.15}$$

where

$$\nabla_j = \begin{bmatrix} \partial J/\partial u(j)_1 \\ \vdots \\ \partial J/\partial u(j)_m \end{bmatrix} \tag{3.16}$$

and ε_{n_j} is chosen to minimize the functional

$$J[\underline{u}(1)^{(n)}, \underline{u}(2)^{(n)}, \cdots, \underline{u}(j)^{(n)} + \varepsilon_{n_j} \nabla_j{}^{(n)}, \cdots, \underline{u}(R)^{(n)}] \tag{3.17}$$

The first iteration is obtained by using the solution of Equation (3.14) as an initial approximation to the control components with any of the

control components that exceeds its bound set equal to the magnitude bound. If none exceeds its bound, then Equation (3.15) gives the same values as obtained from Equation (3.14). The initial values for the second iteration are the results of the first iteration with any controls that exceed their bound set equal to the magnitude bound. The iterative procedure is terminated when the change in each component is less than some specified value. The convergence rate is fast but decreases as the number of control components increases.

States Not Within Prescribed Bounds

Theorem I states that if any point on an optimal trajectory falls in an unallowable region, it should be bounded to the boundary of the allowable region. A procedure for bounding the point directly to the convex boundary is very difficult to implement and, in fact, impossible if no analytical expression is available for the boundary. The purpose of this section is to present an iterative procedure for bounding to the state boundary.

It was shown in the proof of Theorem I that the admissible set of optimal controls with the point bounded to the convex state boundary is also the optimal controls with the point bounded to a hyperplane tangent to the boundary at the optimal bounded point. This suggests that the optimal bounded point on the state boundary can be obtained by iteratively bounding to tangent hyperplanes until the optimal bounded point on the hyperplane coincides with the point of tangency. Because of the convexity and smoothness of the boundary, convergence to any desired accuracy is assured.

The iterative procedure consists of projecting the unacceptable point of the trajectory onto the state boundary, constructing a tangent hyperplane to the boundary at the projection point, and computing the optimal admissible set of controls with the unacceptable point constrained to lie on the tangent hyperplane. If the unacceptable point, when constrained to lie on the hyperplane, falls within a specified distance of the point of tangency, the iterative procedure is terminated. If it does not, this new unacceptable point is projected onto the state boundary and another iteration is performed. Two iterations of this procedure are illustrated by Figure 19.

The effect on the trajectory of bounding a point to the state boundary is illustrated by Figure 20. The bounded trajectory lies inside or closer to the origin of the state space than did the unbounded trajectory. This is caused by the fact that the bounded point itself is pulled in closer to the origin. The cost of the bounded trajectory, however, is always greater than the cost of the unbounded trajectory.

If the saturation boundary is stored in the form of points on the boundary, another procedure may be used. For an nth order system, the n closest points on the S.B. to the unacceptable point on the trajectory are found. These n points describe an nth order hyperplane to which the unacceptable point is bounded with minimum cost. If the bounded point lies on that part of the hyperplane that belongs to the approximated saturation boundary, the iterative procedure is terminated. If it lies outside that region, the closest n points on the S.B. are found and the procedure is repeated. Under certain conditions, this procedure will not converge as will be explained in Chapter IV.

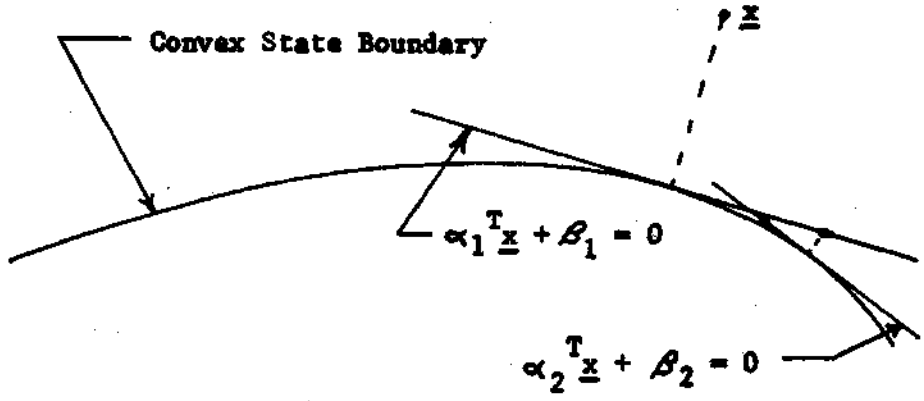


Figure 19. Iterative Procedure for Bounding Points to the State Boundary.

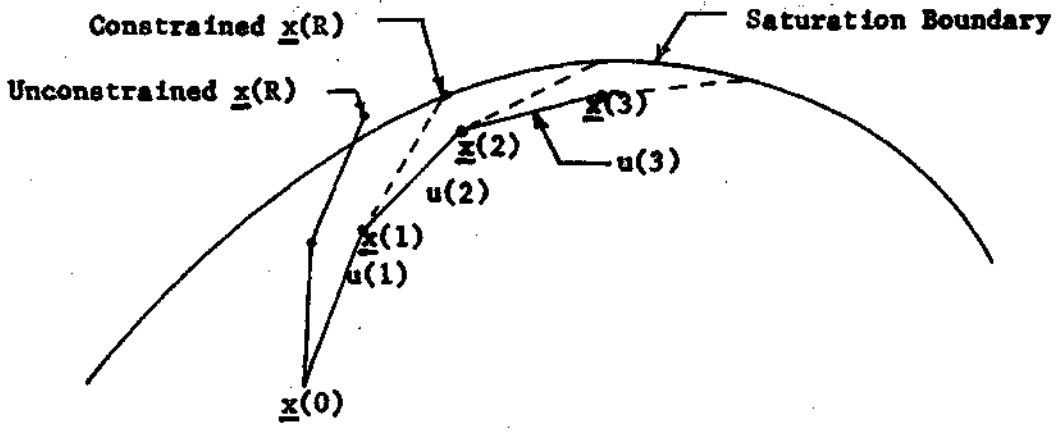


Figure 20. Effect on the Trajectory of Constraining a Point to the Boundary.

The proposed R-step optimal controller is designed so that it first computes the optimal controls with no constraints imposed. The magnitude of the control variables are then checked. If any control component exceeds its bound, the coordinatewise gradient technique is used to obtain the optimal set of admissible controls. Using this set of admissible controls, the trajectory is checked to see if any discrete points are in an unallowable region. If there is a point on the trajectory in an unallowable region, it is bounded to the state boundary as described above. The specific optimization algorithms for R=1 through R=4 are discussed in detail in Appendix I. The technique for expanding the algorithm for larger values of R is also given in Appendix I.

In the R-step optimal controller, the trajectory of the system is modeled over R sampling periods into the future. Since each sampling period is T seconds, the time interval over which the cost is minimized is RT seconds. This RT seconds is referred to as the lead time used in the controller optimization procedure. The effects of this lead time are discussed in detail in Chapter V.

In summary, the proof that points on a trajectory which fall in unallowable regions should be bounded to the boundary of the allowable region was given in this chapter. An R-step optimization algorithm for the bounded control and bounded state problem was presented. The implementation of this algorithm is given in Chapter IV and several examples that illustrate the applicability of this method are given in Chapter V.

CHAPTER IV

IMPLEMENTATION OF THE OPTIMIZATION ALGORITHM

In this chapter, the implementation of the theory of the optimization algorithm is discussed. Generalized expressions for the coefficient matrices in the cost functional are presented and the techniques for finding the unconstrained and the constrained controls are discussed. The techniques for implementing the bounding procedure and the convergence problem encountered when using a piecewise-linear approximation to the saturation boundary are discussed.

General Expressions for the Optimization Algorithm

In this section, general expressions for finding the constant matrices in the cost expression are given. The method for generating the optimal unconstrained controls is discussed and a method for writing the (n+1)th control iteration in the coordinatewise gradient procedure is given.

The general expression for the cost of an R-step optimal controller as a function of the control variables only is given by Equation (3.13). For the quadratic cost considered in this dissertation, Equation (3.13) expands into

$$J = 1/2[\sum_{i=1}^{R} \underline{u}(i)^T C_i \underline{u}(i) + 2 \sum_{j=i+1}^{R} \sum_{i=1}^{R} \underline{u}(i)^T C_a \underline{u}(j) + 2 \sum_{s=1}^{R} C_b \underline{u}(s) + K] \qquad (4.1)$$

where

$$a = j - i + \sum_{k=R-i+1}^{R} k \tag{4.2}$$

and

$$b = s + \sum_{k=1}^{R} k \tag{4.3}$$

The constant matrices all depend on the F, G, P, and Q matrices of the system. These constant matrices for R=1 through R=4 are given in Appendix II and are seen to become more complex as R increases. General expressions for these matrices are given in the following paragraphs.

The constant matrices C_1 through C_R in Equation (4.1) are given by the following expression:

$$C_i = Q + \sum_{k=0}^{R-i} G^T(F^T)^k P(F)^k G \tag{4.4}$$

As an example, the matrix C_1 in a four-step optimal controller is

$$C_1 = Q + \sum_{k=0}^{3} G^T(F^T)^k P(F)^k G \tag{4.5}$$

$$= Q + G^TPG + G^TF^TPFG + G^T(F^T)^2P(F)^2G + G^T(F^T)^3P(F)^3G$$

as given in Appendix II.

The matrices C_a for $1 \leq i \leq R-1$ and for $i < j \leq R$ in Equation (4.1) are given by

$$C_a = \sum_{k=j}^{R} G^T(F^T)^{k-i}P(F)^{k-j}G \tag{4.6}$$

For example, consider the constant matrix for i=2, j=3, in a four-step optimal controller. From Equation (4.2) $a=3-2+\sum_{k=3}^{4} k=8$. Therefore, from Equation (4.6)

$$C_8 = \sum_{k=3}^{4} G^T(F^T)^{k-2}P(F)^{k-3}G \tag{4.7}$$

$$= G^TF^TPG + G^T(F^T)^2PFG$$

as given in Appendix II.

The matrices C_b, where b is given by Equation (4.3) and $s \leq R$, are given by

$$C_b = \sum_{k=s}^{R} (F^T)^k P(F)^{k-s}G \tag{4.8}$$

For example, consider the constant matrix for s=2 in a four-step optimal controller. From Equation (4.3), $b=2+\sum_{k=1}^{4} k=12$. Therefore, from Equation (4.8)

$$C_{12} = \sum_{k=2}^{4} (F^T)^k P(F)^{k-2}G \tag{4.9}$$

$$= (F^T)^2PG + (F^T)^3PFG + (F^T)^4PF^2G$$

as given by the expression for C_{12} in Appendix II.

The expressions for the unconstrained optimal controls may be obtained by taking the partial derivatives of the cost functional with respect to each of the components of the R control variables, equating these resulting R expressions to zero, and solving by Cramer's rule. Appendix II gives these expressions for R=1 through R=4.

The expressions for the (n+1)th iteration of the control vectors in the coordinatewise gradient procedure can be obtained from the cost functional by inspection. The (n+1)th iteration of the ith component of the jth control vector $u(j)_i$ is equal to minus the sum of all of the coefficient matrices of the $u(j)_i$ terms in the cost expression premultiplied by the inverse of the matrix in the $u(j)_i^2$ term. Appendix II gives these expressions for R=1 through R=4.

Implementation of the Bounding Procedure

The general method for optimally bounding a particular point to the convex state boundary was presented in Chapter III. However, no explanation was given as to how this bounding procedure is used in an R-step optimal controller. This section explains the implementation of the bounding procedure when one or more points have to be bounded to the state boundary and also gives the method for unbounding points.

In the optimization algorithm, the point $\underline{x}(R)$ on the optimal state unconstrained trajectory is first determined to be above or below the state boundary as shown in the flow graphs in Appendix I. If $\underline{x}(R)$ is in an unacceptable region, it must be bounded to the state boundary by the method outlined in Chapter III. The control $\underline{u}(R)$ is used to constrain the point $\underline{x}(R)$ to hyperplanes tangent to the state boundary until

$\underline{x}(R)$ is within an acceptable distance from the point of tangency. Equation (3.2) is solved for $u(R)_p$ and substituted into the cost functional. The cost, which is now a quadratic functional in $\underline{u}(1)$ through $\underline{u}(R)'$, is minimized with an admissible set of controls by using the coordinatewise gradient procedure. Then $u(R)_p$, which is a function of $\underline{u}(1)$ through $\underline{u}(R)'$, is found. If $|u(R)_p| > U_{max}$, then another component of $\underline{u}(R)$ must be constrained to force $\underline{x}(R)$ to lie on the state boundary. This procedure is repeated until the constrained control is an acceptable control. Since the initial point $\underline{x}(0)$ is on or below the state boundary, not all of the controls required to bound $\underline{x}(R)$ to the boundary are greater than U_{max}. This fact assures an acceptable solution.

If, in the test procedure, a point other than $\underline{x}(R)$ on the computed trajectory lies in an unacceptable region, the bounding technique becomes slightly more complicated. Suppose that the point $\underline{x}(R-j)$, where $i \leq j \leq R-1$, is found to lie in an unacceptable region. Then $\underline{x}(R-j)$ must be bounded to the state boundary by the same technique as used to bound $\underline{x}(R)$. In this procedure, the points $\underline{x}(R-j+1)$ through $\underline{x}(R)$ are unconstrained and the controls $\underline{u}(R-j+1)$ through $\underline{u}(R)$ have no effect on the location of the point $\underline{x}(R-j)$. The magnitude constraint on the control variables is handled in the same manner as in the $\underline{x}(R)$ bounding procedure. When an acceptable solution is found, the point $\underline{x}(R-j+1)$ is checked for acceptability. If $\underline{x}(R-j+1)$ is in an unacceptable region, the points $\underline{x}(R-j)$ and $\underline{x}(R-j+1)$ are constrained to lie on the state boundary by constraining $\underline{u}(R-j)$ and $\underline{u}(R-j+1)$. Of course, the magnitude

of $\underline{u}(R-j+1)$ is equal to U_{max} in this procedure. This procedure is repeated until all of the points $\underline{x}(R-j)$ through $\underline{x}(R)$ are constrained to the state boundary or until a point $\underline{x}(R-k)$, where $k < j$, is found to lie in an acceptable region without being bounded. If a point $\underline{x}(R-k)$ is found which lies in an acceptable region without being bounded, then because of the convexity of the trajectory and of the state boundary, it follows that all the points $\underline{x}(R-k)$ through $\underline{x}(R)$ are acceptable points. The procedure outlined in this paragraph is general in that it handles both the bounding and unbounding of points on the trajectory in an optimal manner.

Convergence Problem

As pointed out in Chapter III, there exist special cases in which the iterative procedure for bounding a point on the optimal trajectory to a piecewise-linear approximation of the state boundary will not converge. These special cases are investigated in this section and a necessary condition to obtain convergence to the piecewise-linear boundary is given.

In order to obtain a necessary condition that will assure convergence, the concept of reachable regions in reference (5) is employed. The reachable region is defined to be the locus of all points which can be reached from any given initial point or initial region in the state space with the control available. The order of the reachable region is defined to be the number of independent orthogonal axes necessary to describe the region.

Consider an nth order linear plant with m control variables described by the linear difference equation

$$\underline{x}(k) = F\underline{x}(k-1) + G\underline{u}(k) \tag{4.10}$$

If m=0, then the trajectory is given by

$$\underline{x}(k) = F\underline{x}(k-1) \tag{4.11}$$

Therefore, from any initial point $\underline{x}(0)$ in state space, the locus of all possible points $\underline{x}(1)$ on the trajectory after one sampling period is a single point in state space. In fact, the reachable region after R sampling periods is zero order if m=0. If m=1, then the reachable region after one period is a straight line as shown in the second order example in Figure 21. The reachable region in this case is said to be of first order. After two sampling periods, the reachable region from the point $\underline{x}(0)$ is second order as shown in the second order example in Figure 22. Therefore, if m=1, the order of the reachable region is equal to R if $R < n$ and is equal to n if $R \geq n$. The order of the reachable region cannot be greater than the order of the system.

From the results of the above paragraph, it is obvious that if the number of control variables in the system is m, then the order of the reachable region after one sampling period is m and after R sampling periods is Rm if $Rm < n$ or is n if $Rm \geq n$. Therefore, a necessary condition for the order of the reachable region to be equal to the order of the plant is

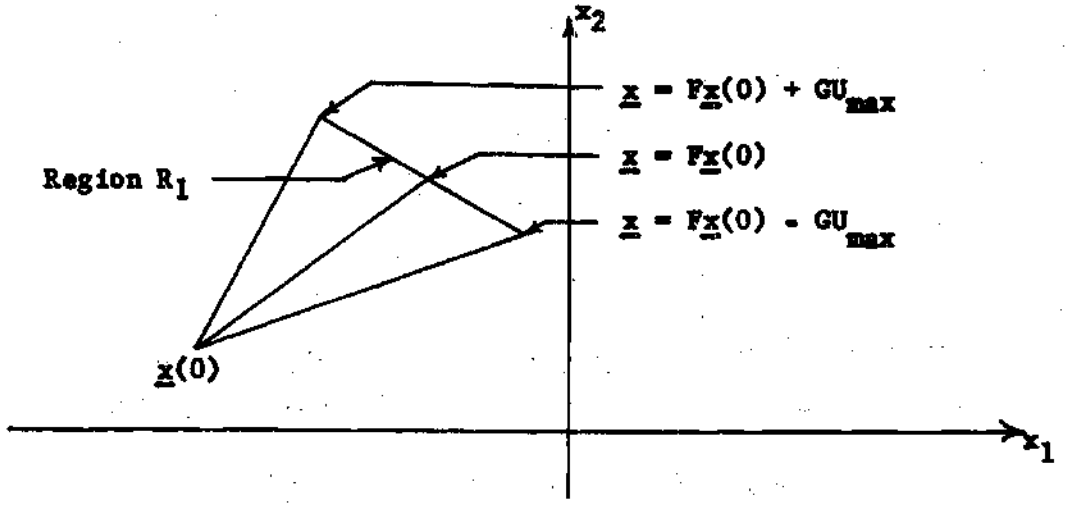


Figure 21. The Construction of the Region R_1.

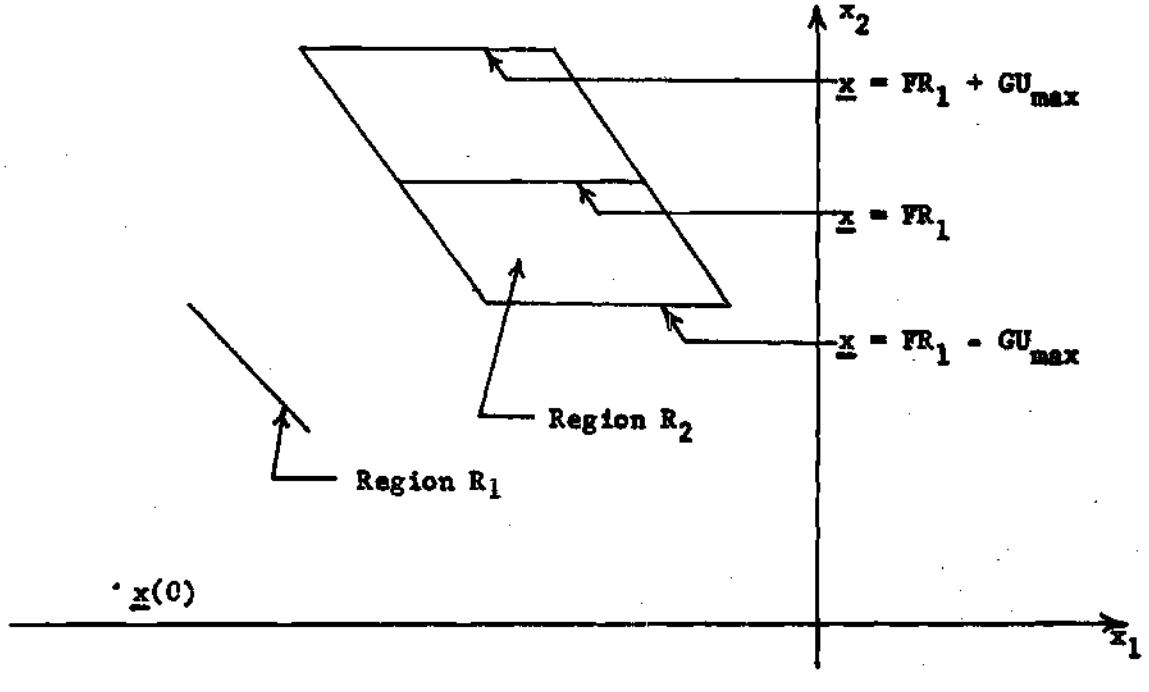


Figure 22. The Construction of the Region R_2.

$$R \geq I \tag{4.12}$$

where I is the smallest integer greater than or equal to n/m.

Consider first, the case where the reachable region is of order n and the state boundary is approximated with nth order hyperplanes. If a hyperplane can be found to which the unacceptable point, when bounded, lies on that part of the hyperplane that belongs to the approximated boundary, then the iterative procedure is said to have converged. Figure 23 illustrates this possibility for a second order example. From the proof of Theorem I, minimum cost is obtained when the unacceptable point is bounded to the point of tangency of the hyperplane with a constant cost contour.

There exists the possibility that no hyperplane can be found to satisfy the requirements in the paragraph above. Suppose that the unacceptable point is first constrained to hyperplane A and lies above the state boundary when the cost is minimized. The unacceptable point is then constrained to lie on the closest hyperplane B and again falls above the state boundary when the cost is minimized. If the closest hyperplane is again hyperplane A, then the iterative procedure will oscillate between hyperplanes A and B and will not converge. From the proof of Theorem I, however, the cost monotonically increases as the unacceptable point is forced away from the points yielding minimum cost on hyperplanes A and B. It follows that the point yielding minimum cost and satisfying the state constraint must lie somewhere along the intersection of the two hyperplanes. Since the initial point of the trajectory is on or below the boundary approximation, at least one point below

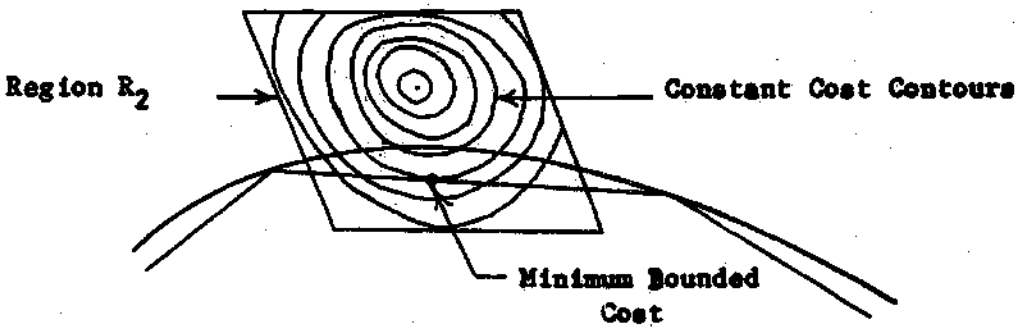


Figure 23. Unique Solution with Approximated Boundary.

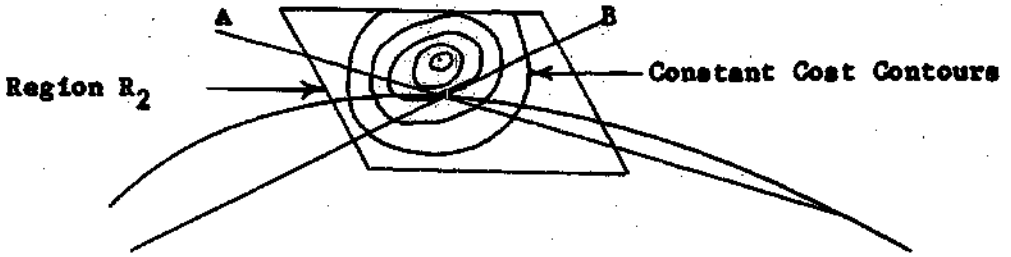


Figure 24. Non-Convergent Solution with Approximated Boundary.

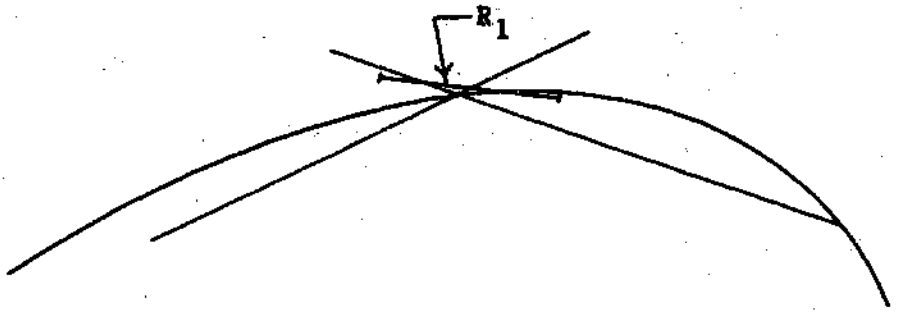


Figure 25. Non-Convergent Solution with Approximated Boundary.

the state boundary on hyperplane A or B is reachable. The reachable region, however, is convex as proven in reference (5). It follows that the intersection of hyperplanes A and B is reachable. A procedure must be placed in the algorithm to recognize this non-convergence and to find the minimum cost under these conditions. Figure 24 illustrates this non-convergence for a second order system.

When the order of the reachable region is less than the order of the plant, the convergence problem discussed in the above paragraph cannot be handled in some cases. This problem arises when the intersection of hyperplanes A and B are not in the reachable region. Figure 25 illustrates this case for a second order system with a first order reachable region. It is seen that the reachable region does intersect the state boundary but does not intersect the approximation to the state boundary. Although non-convergence may not occur when the order of the reachable region is less than the order of the plant, the possibility exists and, therefore, this situation should be avoided by choosing $R \geq I$.

In summary, a simple and compact method for obtaining the coefficient matrices in the cost expression, and the expressions for the unconstrained and constrained control variables was given in the first section of this chapter. The implementation of the bounding procedure was discussed in the second section, and the convergence problem when using a piecewise-linear approximation to the saturation boundary was discussed in the final section. This chapter completes the theory and implementation technique for the optimization algorithm capable of

handling discrete systems with bounded control and bounded state variables.

CHAPTER V

ANALYSIS OF THE CHARACTERISTICS OF THE OPTIMIZATION ALGORITHM USING EXAMPLE PROBLEMS

In the preceding chapters, an algorithm was developed to control discrete-input systems with bounded state and bounded control variables. Two examples are presented in this chapter which illustrate the applicability of the method. Both examples are second order for ease and simplicity of presentation. The first example illustrates the mathematical procedures used in the optimization algorithm. The second example is the most general and is the one used to analyze the characteristics of the optimization algorithm.

Example I

The purpose of this example is to illustrate some of the mathematical procedures used in the optimization algorithm. An analysis of the results obtained using the optimization algorithm will be presented in Example II using a more general plant which has complex roots.

Consider the first example in Chapter II which is redefined in this section for convenience. The plant to be controlled is described by the differential equations

$$\dot{\underline{x}} = A\underline{x} + Bu$$

or

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0.0 & 1.0 \\ 0.0 & -1.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix} u \qquad (5.1)$$

Using a sampling period of one second, the discrete equations for the plant described by Equation (5.1) are

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} 1.0 & 0.632 \\ 0.0 & 0.368 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.368 \\ 0.632 \end{bmatrix} u(k) \qquad (5.2)$$

The cost functional is

$$J = 1/2 \sum_{k=1}^{2} [20.0x_1(k)^2 + x_2(k)^2 + u(k)^2] \qquad (5.3)$$

The constraint on the states of the system is

$$\underline{\alpha}^T \underline{x} + \beta \leq 0$$

or

$$2.0x_1(k) + 1.0x_2(k) - 1.0 \leq 0 \qquad (5.4)$$

while the constraint on the control variables is

$$|u(k)| \leq U_{max} = 1.0, \qquad k = 1,2 \qquad (5.5)$$

The first step in the determination of the optimization algorithm is the determination of the saturation boundaries. The point of tangency is found from Equations (2.10) and (2.11). Substituting Equations (5.1), (5.3), and (5.4) into Equations (2.10) and (2.11) yields

$$\begin{aligned} \underline{\alpha} \cdot (A\underline{x} + Bu) &= 0 \\ \underline{\alpha}^T \underline{x} + \beta &= 0 \end{aligned} \tag{5.6}$$

Solving Equations (5.6) simultaneously yields $x_1=0$, $x_2=+1$ as shown in Figure 26. Since the plant is time-invariant and has real roots only, the method described in Chapter II, using Equation (2.18), may be used to determine an analytical expression for the saturation boundary as

$$x_1 + x_2 - \ell n[1/2(x_2+1)] - 1 = 0 \tag{5.7}$$

The S.B. given by Equation (5.7) is shown in Figure 26.

If the discrete equations are used to determine the point of tangency for the S.B. using Equations (2.12), (2.13), and (2.14), the resulting two points are $x_1=-0.58$, $x_2=2.16$ and $x_1=0.42$, $x_2=0.16$. The point of tangency, found by using the continuous equations of the plant, lies between the two points found by using the discrete equations of the plant as pointed out in Chapter II.

By successive substitution of the difference equations into Equation (5.3), the cost may be reduced to the following function of the control variables and initial conditions only:

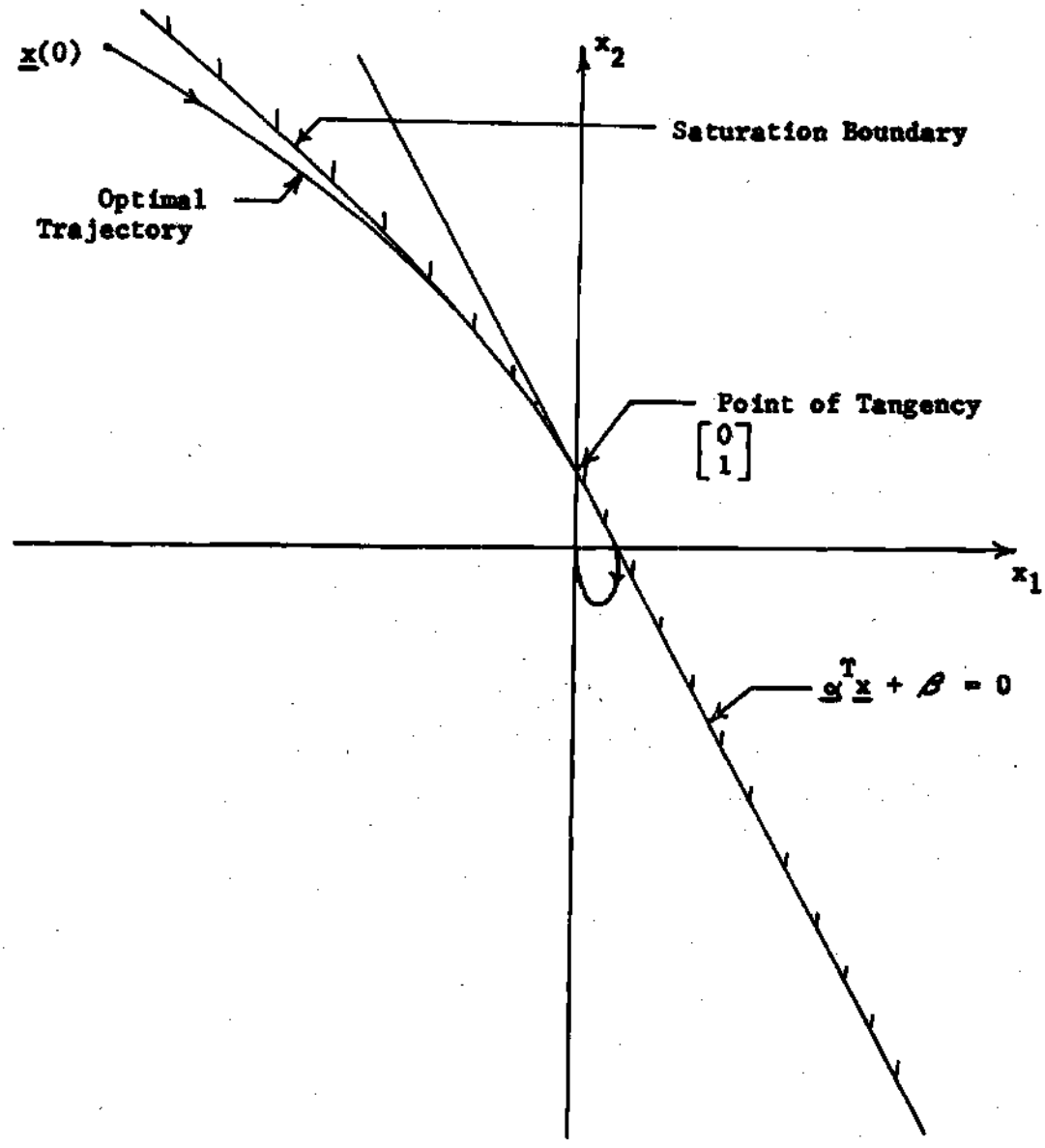


Figure 26. Optimal Trajectory for System with Real Roots.

$$J = 15.94\ u(1)^2 + 4.11\ u(2)^2 + 11.59\ u(1)u(2) + 45.41\ u(1)x_1(0) \quad (5.8)$$

$$+ 36.37\ u(1)x_2(0) + 14.72\ u(2)x_1(0) + 12.89\ u(2)x_2(0) + k$$

The unconstrained controls may be obtained from Equation (3.14) and are

$$u(1) = -1.587\ x_1(0) - 1.170\ x_2(0)$$

$$u(2) = \ 0.448\ x_1(0) + 0.081\ x_2(0) \quad (5.9)$$

The expressions for the control variables in the coordinatewise gradient iterative procedure are obtained from Equation (3.15), (3.16), and (3.17) and are

$$u(1)^{(n+1)} = -0.363\ u(2)^{(n)} - 1.425\ x_1(0) - 1.141\ x_2(0)$$

$$u(2)^{(n+1)} = -1.411\ u(1)^{(n)} - 1.792\ x_1(0) - 1.570\ x_2(0) \quad (5.10)$$

As proven in Chapter III, if a point on the trajectory using either the control variables of Equation (5.9) or (5.10) lies above the convex state boundary, it should be bounded to the state boundary. For example, assume that the Rth point, $\underset{\sim}{x}(2)$, lies above the state boundary. In this example, the unacceptable point is projected along a constant x_1 axis onto the boundary and a line tangent to the boundary is constructed at this point as shown in Figure 19 (p. 47). The equation of this line is

$$x_1 + SL\, x_2 + \beta = 0 \qquad (5.11)$$

where SL is evaluated from the slope of the tangent line and β is related to the x_2 intercept. The point $\underline{x}(2)$ may be constrained to the tangent line by constraining u(2) as follows:

$$u(2) = C1\, u(1) + C2$$

where

$$C1 = -(0.233SL + 0.768)/(0.632\ SL + 0.368)$$

and

$$C2 = -[x_1(0) + (0.135\ SL + 0.865)x_2(0) + \beta]/(0.632\ SL + 0.368) \qquad (5.12)$$

Substituting this expression for u(2) into the cost expression of Equation (5.8) and minimizing yields

$$u(1) = -[(8.22\ C1 + 11.6)C2 + (14.72\ C1 + 45.4)x_1(0) + (12.92\ C1 + 36.3)x_2(0)]/ \qquad (5.13)$$

$$(8.22\ C1^2 + 23.2\ C1 + 31.92)$$

If the magnitude of u(1) is larger than U_{max}, u(1) is set equal to $\pm\ U_{max}$. If the control u(2) computed using Equation (5.12) is greater in magnitude than U_{max}, u(1) must be used to constrain $\underline{x}(2)$ to the

tangent line. Since $\underline{x}(0)$ is below the boundary, both u(1) and u(2) will not be greater in magnitude than U_{max} which assures an acceptable solution.

An optimal bounded trajectory for this system is shown in Figure 26 (p. 64). If the algorithm for constraining the states described above were not used, the trajectory would violate the state boundary. The optimal trajectory, which is tabulated in Table 1, first becomes tangent to the S.B. and then is coincident with the S.B. and the original boundary before breaking away toward the origin. The problem is considered solved when the trajectory enters a square $|x_1| \leq 0.05$ and $|x_2| \leq 0.05$.

Table 1. Optimal Control and Trajectory for System in Example 1

Step No.	Control	Step Cost	Total Cost	x_1	x_2	States Constrained(C) Unconstrained(U)
0	-	-	-	-47.00	50.00	-
1	0.28152	5024.82	5024.82	-15.30	18.58	C
2	-0.98654	346.62	5371.44	-3.92	6.21	C
3	-1.00000	6.32	5377.76	-0.36	1.65	C
4	-1.00000	3.02	5380.79	0.32	-0.02	U
5	-0.47709	0.65	5381.44	0.13	-0.31	U
6	0.15803	0.03	5381.46	-0.01	-0.014	U

Example II

Consider the system shown in Figure 5 (p. 15) where the plant to be controlled is described by the differential equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0.0 & 1.0 \\ -1.0 & 1.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix} u \qquad (5.14)$$

Since several different sampling periods are used with this system for comparison purposes, no discrete equations are given. The cost functional is

$$J = 1/2 \sum_{k=1}^{R} [x_1(k)^2 + x_2(k)^2 + 10.0\ u(k)^2] \qquad (5.15)$$

where R is allowed to vary between one and four for comparison purposes. The state constraint is

$$x_1(k) + 2.0\ x_2(k) - 2.0 \leq 0 \qquad (5.16)$$

and the constraint on the control variable is

$$|u(k)| \leq U_{max} = 1.0, \quad k = 1,2,\cdots,R \qquad (5.17)$$

The expressions for the cost functionals, unconstrained optimal controls, and magnitude constrained optimal controls in the coordinate-wise gradient procedure for R=1 through R=4 are given in Appendix II.

No numerical values are given since all the values depend on the F and G matrices which depend on the sampling period used.

The point of tangency of the saturation boundary with the original boundary is x_1=-2.0 and x_2=+2.0. Because of the transcendental nature of the equations, no analytical expression for the S.B. can be obtained. The approximation technique discussed in Chapter II is used. The points on the S.B. that are used to approximate the boundary are found by integrating the difference equations for the system backwards in time from the point of tangency as indicated by Equation (2.15). In order to obtain an accurate approximation to the S.B., the sampling period used to generate the points should be small compared with the sampling period of the system. However, as the sampling period used to generate the S.B. is decreased, the amount of storage space required in the controller is increased and the computation time is increased. Eventually there has to be a trade-off between the desired accuracy and the storage space and computation time required. Figure 27 gives a comparison of approximations to the saturation boundary using generation sampling periods of 1.0, 0.5, and 0.1 seconds. A sampling period of 0.1 seconds is used in this example to generate the approximation to the S.B. except in those cases where a comparison is made of the effects of varying the sampling period.

The boundary approximation for this second order example may be implemented in the controller in the following manner. Points on a computed trajectory that lie outside of the approximated boundary are bounded to the linear segments described by the two closest points on the boundary. If a point, when bounded, does not lie on that part of

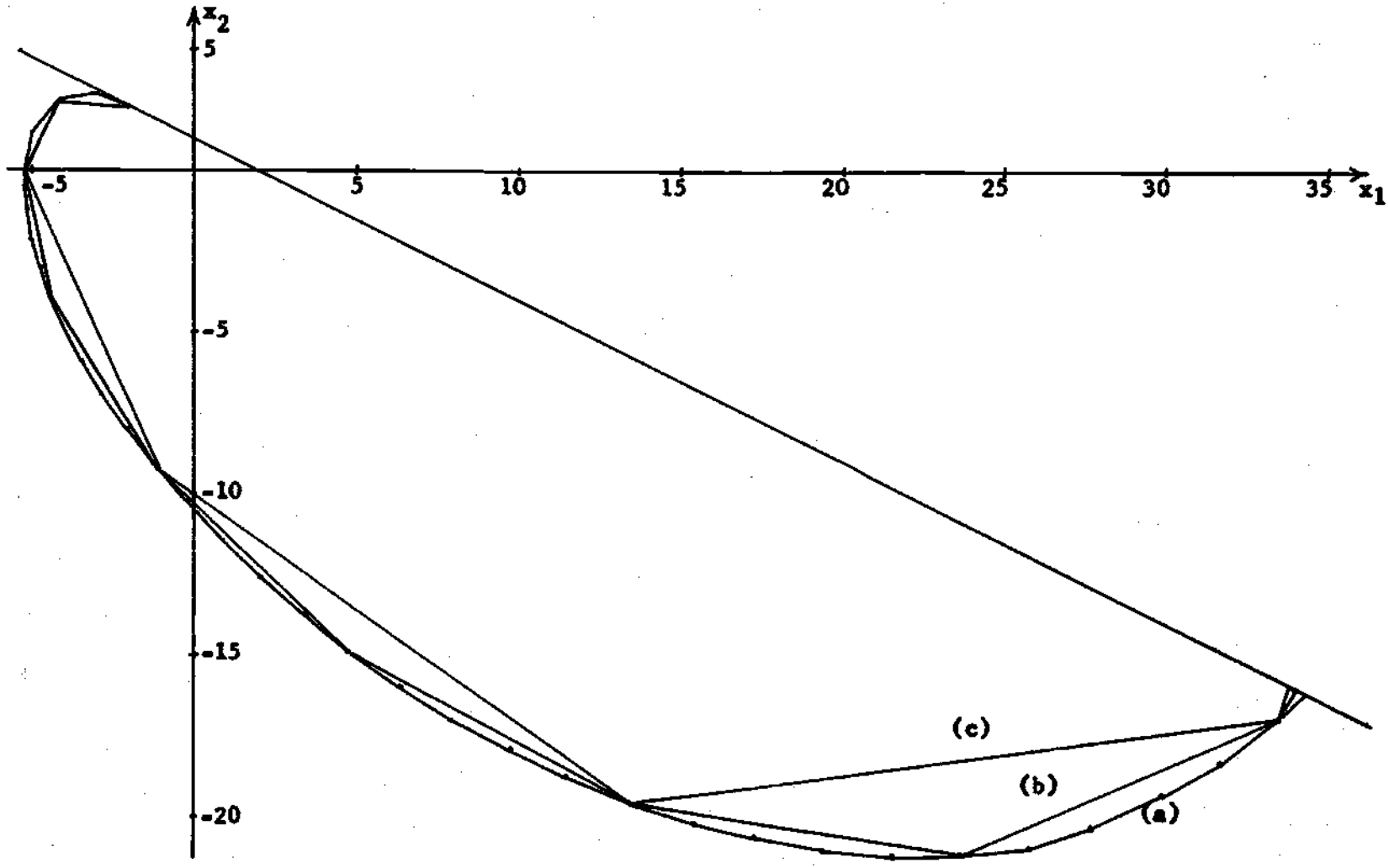


Figure 27. Comparison of Saturation Boundary Approximation for Sampling Periods of (a) 0.1 sec., (b) 0.5 sec., (c) 1.0 sec.

the linear curve which is a part of the boundary, the procedure is repeated. The convergence problem, which may arise when using this procedure, can be handled by a special procedure as explained in Chapter IV.

Comparison of Results

The initial point for all of the trajectories in the comparisons that follow is x_1=30.0 and x_2=-18.0. This initial point was chosen because from this point all trajectories have to be bounded to assure an acceptable solution. Bounding is necessary in order to make valid comparisons.

Comparison of Cost vs. Approximation Boundary Sampling Period

A comparison of total cost of the optimal trajectory vs. the sampling period used to generate the approximation to the S.B. is given in this section. Consider the approximate boundaries with sampling periods of 0.1, 0.2, 0.3, and 0.5 seconds shown in Figure 28. Suppose that the point $\underline{x}$ is a state unconstrained point on a computed trajectory which lies in an unacceptable region. The points on the approximate boundaries which yield minimum cost when the point $\underline{x}$ is bounded to the respective boundaries are shown. The convex constant cost contours are shown tangent to the linear segments at the minimum cost points. Since, as proven in Theorem I, the cost of these contours monotonically increases with increasing distance from the state unconstrained point $\underline{x}$, it follows that

$$J[S.P.=0.1] < J[S.P.=0.5] < J[S.P.=0.2] < J[S.P.=0.3] \qquad (5.18)$$

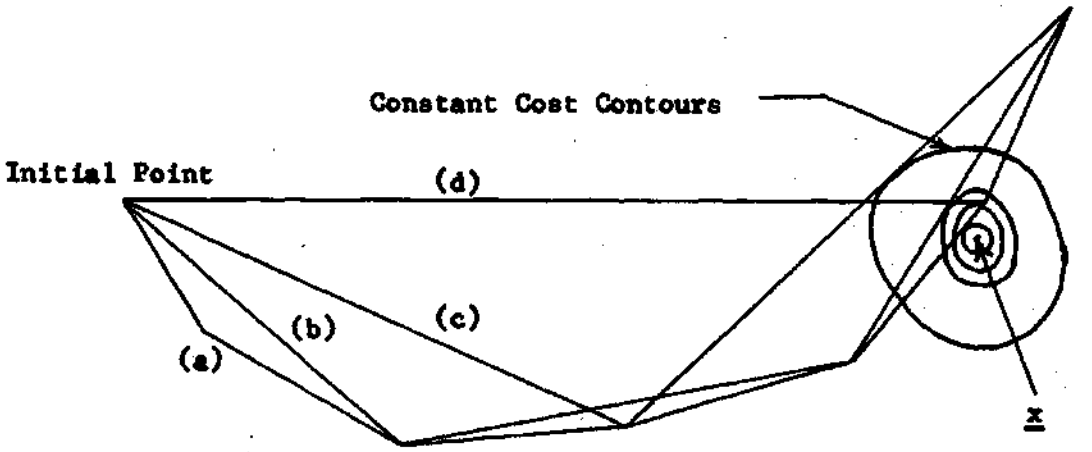


Figure 28. Comparison of Approximate Boundaries Using Sampling Periods of (a) 0.1 sec., (b) 0.2 sec., (c) 0.3 sec., (d) 0.5 sec.

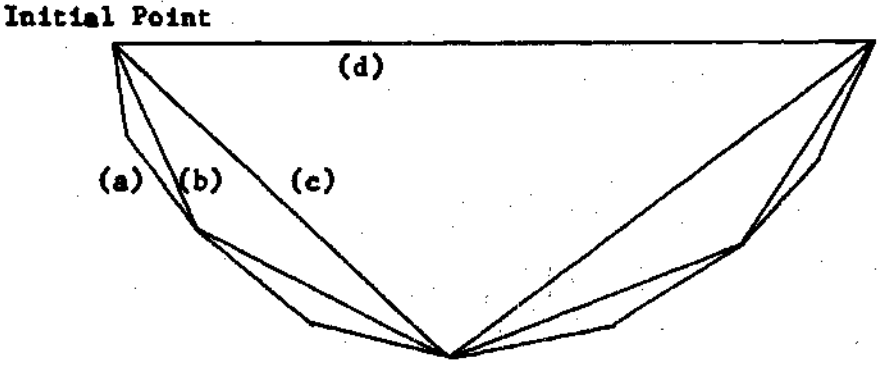


Figure 29. Approximate Boundaries Generated with Sampling Periods of (a) 0.1 sec., (b) 0.2 sec., (c) 0.4 sec., (d) 0.8 sec.

where J[S.P.=0.1] is the cost of the step when the unacceptable point is bounded to the approximate boundary generated by using a sampling period of 0.1 seconds. Since the cost of each step decreases very rapidly as the trajectory approaches the origin as illustrated in Table 1 (p. 67) of Example I, the inequality of Equation (5.18), if it occurs at one of the initial steps of the trajectory, might remain the same when comparing the total costs of the four bounded trajectories. Therefore, a comparison of the total cost of bounded trajectories vs. the sampling periods of the approximation boundaries in Figure 28 (p. 72) is meaningless.

Table 2. Comparison of Total Cost vs. Sampling Period Used to Generate the Approximation to the Saturation Boundary

Sampling Period for Generation of Saturation Boundary	Total Cost of Trajectory
0.1 Second	575.02420
0.5 Second	575.33225
1.0 Second	587.12441

A meaningful comparison may be obtained, however, if each sampling period chosen is an integer multiple of every smaller period chosen, such as 0.1, 0.2, 0.4, and 1.2 seconds. A sampling period of 1.0 second could not be used since 1.0 is not an integer multiple of 0.4. This selection of sampling periods assures that each approximate boundary

lies completely inside all approximate boundaries with smaller generating sampling periods as illustrated in Figure 29 (p. 72). This also assures the same result for the bounded trajectories using these approximate boundaries. Table 2 gives a comparison of the total cost vs. the sampling period used to generate the approximate boundaries for the system in this example with a sampling period of 1.0 second. As expected, the total cost of the trajectory increases as the approximation to the saturation boundary becomes more coarse.

Comparison of Constrained and Unconstrained Trajectories

Figure 30 gives a comparison of the state unconstrained and constrained trajectories for this second order example with R=1 and a sampling period of one second. The trajectories are plotted as smooth curves rather than straight line approximations. During the first sampling interval, the two trajectories are identical since the state boundary is not violated. The trajectories are not the same during the second sampling interval, however, since the unconstrained trajectory violates the saturation boundary. As explained in the theory in Chapter III, the state constrained trajectory lies inside the unconstrained trajectory and has a greater cost. This increased cost must be accepted, however, in order to obtain an acceptable solution.

Analysis and Comparisons Based on R

Next, the effect of increasing R on the trajectory and on the cost is investigated. There are two ways to obtain a comparison as R is varied. The first is to keep the sampling period constant, thereby varying the lead time used in the computations as R is varied. The second is to keep the lead time constant by varying the sampling period

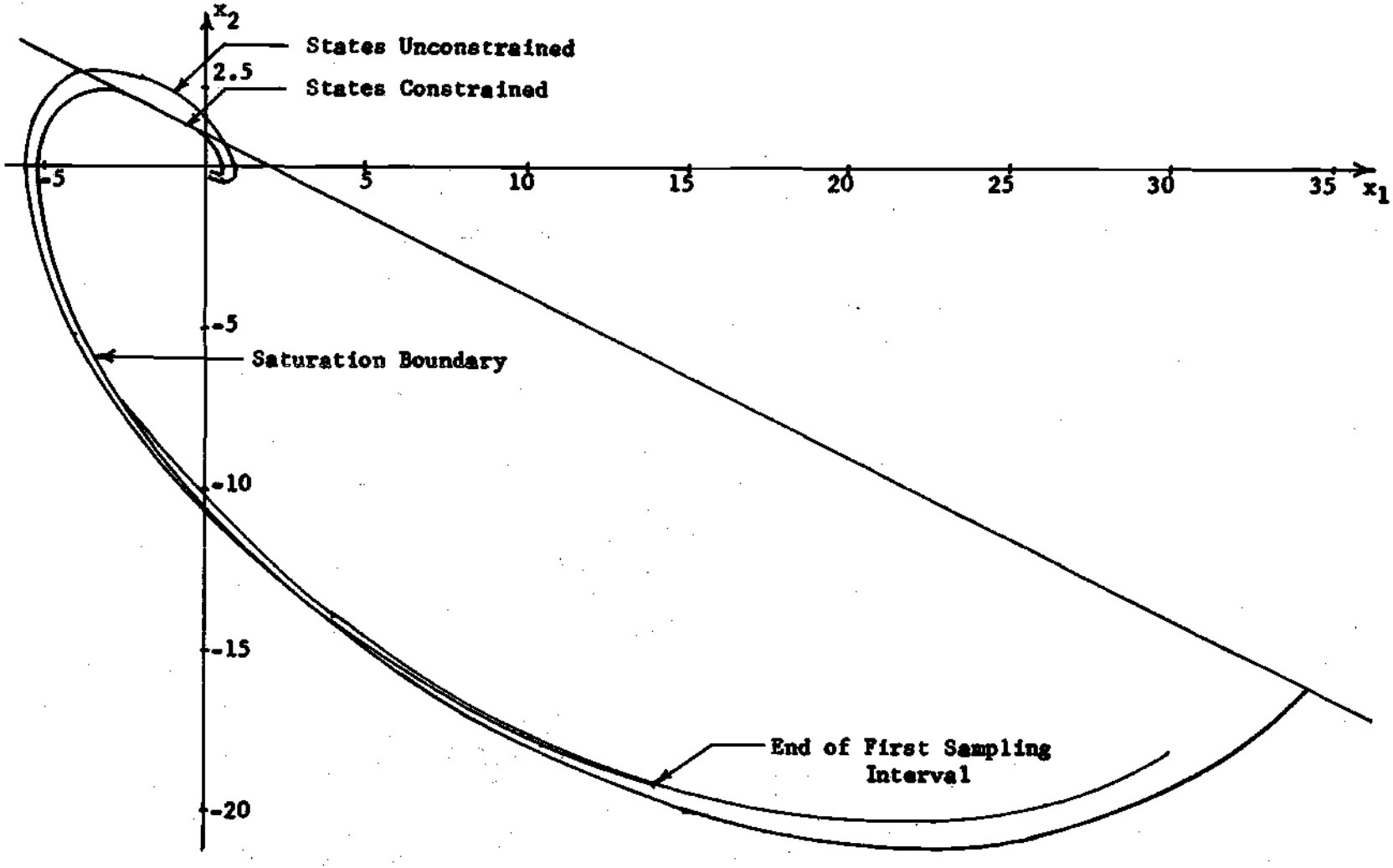


Figure 30. Comparison of Constrained and Unconstrained Trajectories.

as R is varied. To determine the effect of the lead time used in the calculations, consider the linear continuous system with no constraints. The computation of the feedback laws for this problem involves the solution of a two-point, boundary-value problem. These feedback laws, obtained from the solution of the resulting Riccati equation, are exponential as shown in Figure 31. The settling time is about three seconds. Since the lead time required to obtain a steady state control law for the discrete system is related to the settling time of the Riccati equation, the effects of a variable lead time in the comparisons may be minimized by making the smallest lead time greater than the settling time of the Riccati equation. Figure 32, which gives the total cost vs. the lead time, illustrates that no significant improvement in the cost can be obtained for the discrete unconstrained problem by taking a lead time greater than three seconds.

Figure 32 also gives a plot of the total cost vs. lead time for the state constrained problem. As expected, the cost of the constrained problem is greater than the cost for the unconstrained problem. The two curves are similar although the cost for the constrained problem does not approach a constant value as fast as the unconstrained problem. The reason is that in addition to the effects of the settling time of the control laws, the projected violation of the state boundary also effects the trajectory and therefore the cost. As R is increased, the state boundary violation is foreseen sooner, which allows the trajectory to be readjusted sooner. This results in a trajectory with smaller cost. This additional reduction in cost as R is increased depends on the initial condition and cannot be predicted in advance.

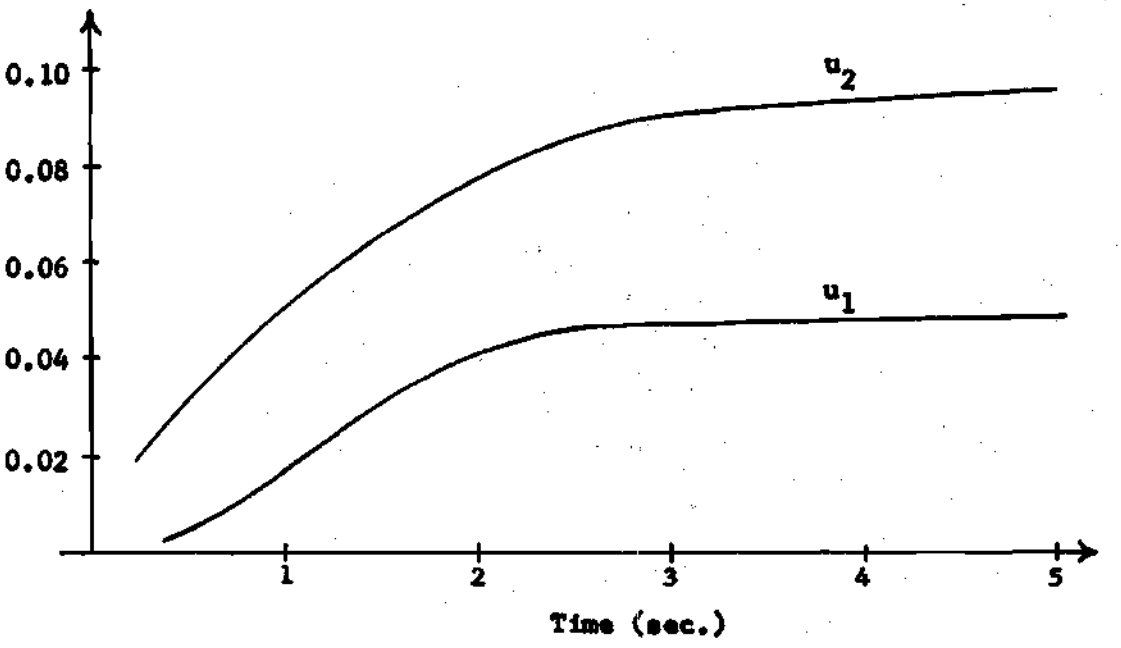


Figure 31. Feedback Laws from Riccati Equation.

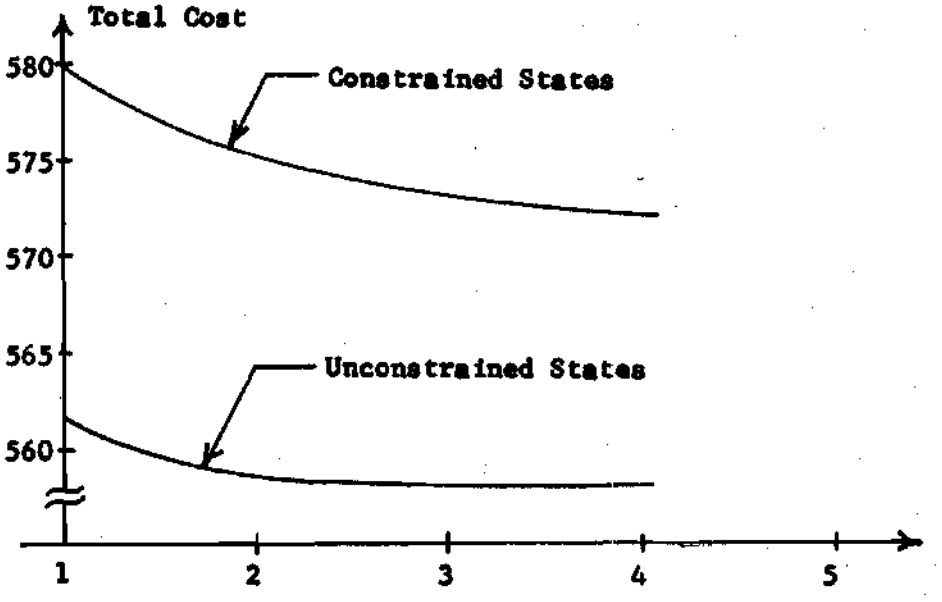


Figure 32. Total Cost vs Lead Time for the Unconstrained Problem and the Constrained Problem.

The effects of the lead time on the cost in this example can be minimized by using a constant lead time as R is varied or by using a variable lead time over three seconds. Since some of the lead times used in the comparisons to follow are of necessity less than three seconds, a constant lead time is used in the comparisons.

A comparison of the constrained trajectories for R=1 and R=3 is given in Figure 33. A constant lead time of 1.2 seconds is used to eliminate the effects discussed in the above paragraphs. Therefore, the sampling period for the R=1 trajectory is 1.2 seconds and the sampling period for the R=3 trajectory is 0.4 seconds. Since the control applied to the system with R=3 is readjusted three times as often as with R=1, the trajectory for R=3 is expected to cost less than the trajectory for R=1. The plot in Figure 34 (p. 80) of total cost vs. R for a lead time of 1.2 seconds confirms this result. For a fixed lead time, as R is increased, the sampling period decreases and the total cost approaches the cost of the continuous constrained solution using the same lead time. Figure 35 (p. 80) is another plot of the total cost vs. R with a lead time of 1.6 seconds. In Figures 34 and 35 (p. 80), if the cost were evaluated at the sampling instants which do not occur in equal increments of time, no valid comparison would result because of the quadratic nature of the cost functional (i.e., one-half of the cost of the trajectory with a sampling period of 0.2 seconds cannot be compared with the cost of the trajectory with a sampling period of 0.4 seconds). Therefore, in order to obtain a valid comparison of total cost in Figures 34 and 35, the cost for all trajectories is evaluated at 0.2 second intervals.

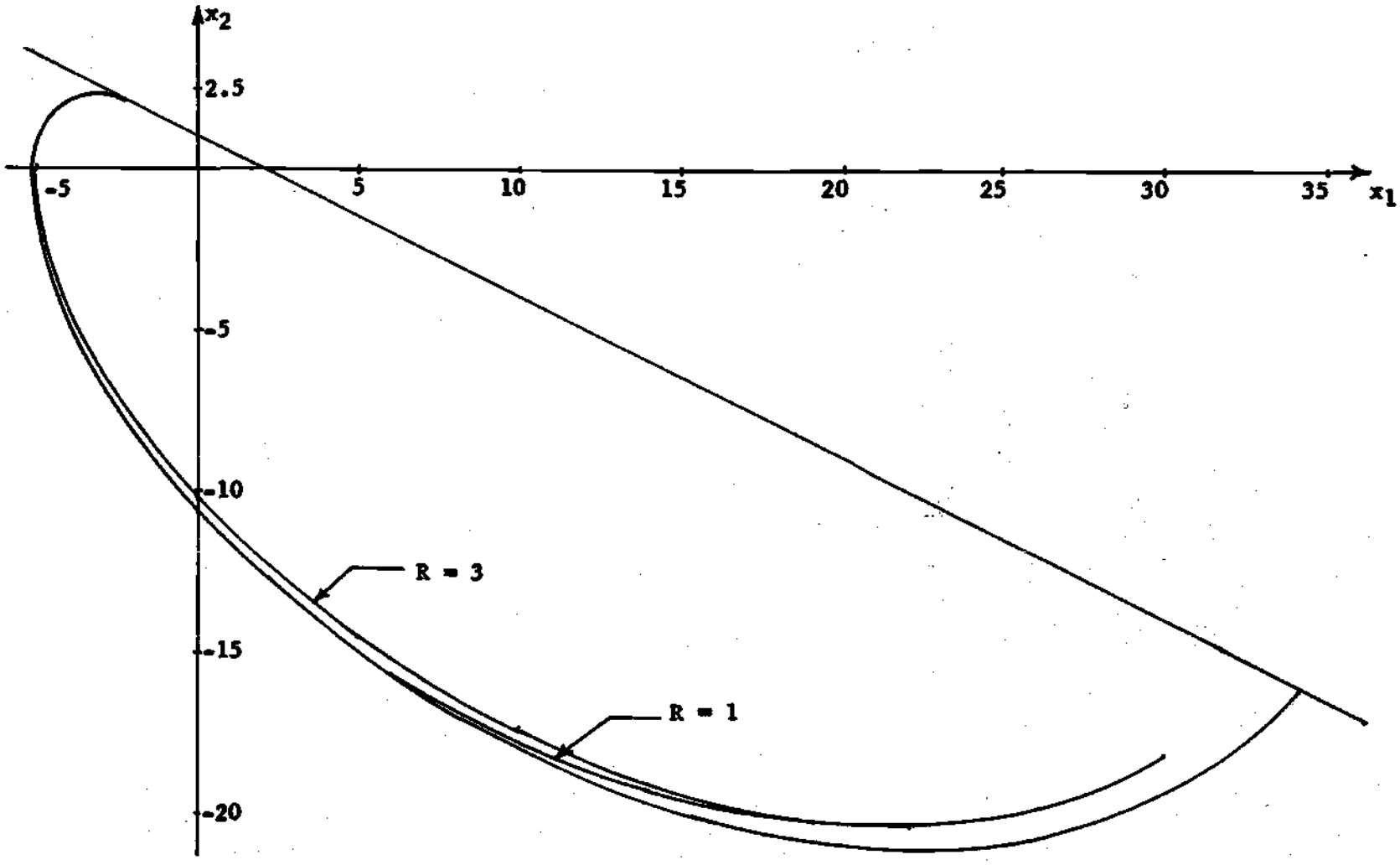


Figure 33. Comparison of Trajectories for R = 1 and R = 3.

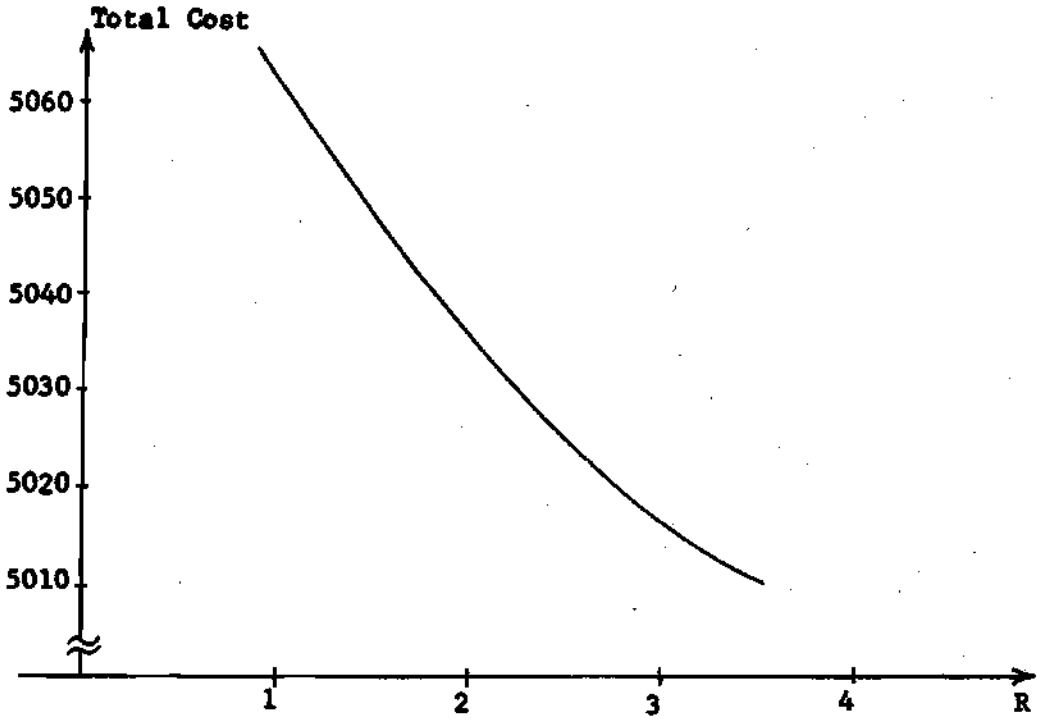


Figure 34. Total Cost vs. R for a Lead Time of 1.2 Seconds.

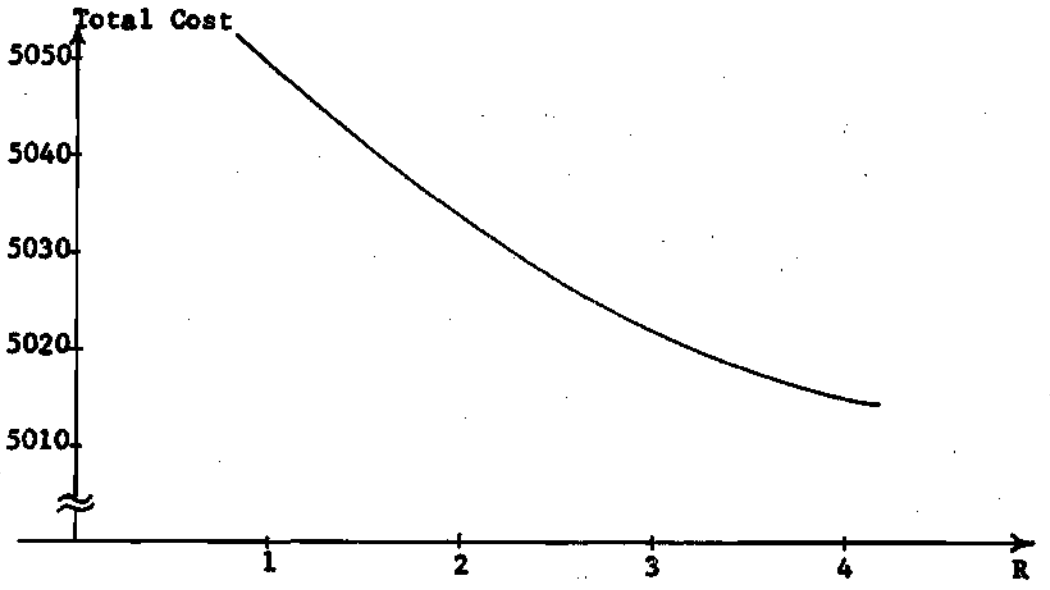


Figure 35. Total Cost vs. R for a Lead Time of 1.6 Seconds.

The effect of holding R fixed (R=4) and varying the sampling period of the system is shown in Figure 36 (p. 82). There are two factors influencing the cost of the trajectory as the sampling period is varied. First, as the sampling period is increased, the lead time is increased and a corresponding reduction in cost is obtained because of the settling of the control laws as indicated in Figures 31 and 32 (p. 77). This factor has very little effect on the cost for sampling periods greater than 0.6 seconds since the settling time for this system is approximately three seconds. The second factor influencing the cost is the effect of applying constant controls to the system over longer periods as the sampling period is increased. As explained in a previous paragraph, this results in a higher cost trajectory. The cost of the trajectory due to this effect, however, approaches a constant value as the sampling period approaches zero. The net effect of these two factors is an increase in cost for small and large values of sampling periods as shown in Figure 36. Therefore, if the system is to be designed with a given R, the sampling period of the system should be chosen to minimize the two effects explained above. For the system in Example II, the sampling period should be approximately 0.5 or 0.6 seconds when R is four.

Computation Time Required

One final comparison is the average time required per computation versus R. As can be seen from the figures in Appendix I, the optimization algorithm becomes more complex as R is increased. The average time required for each computation is therefore expected to increase as R is increased. This increase in average computation time per step is

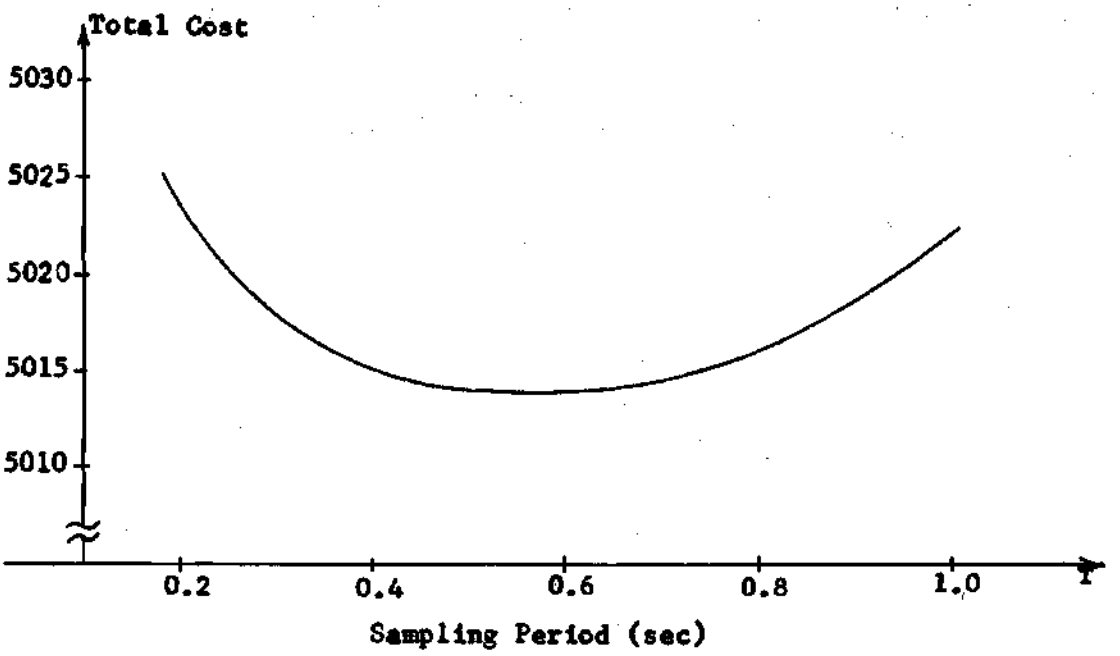


Figure 36. Total Cost vs. Sampling Period with R = 4.

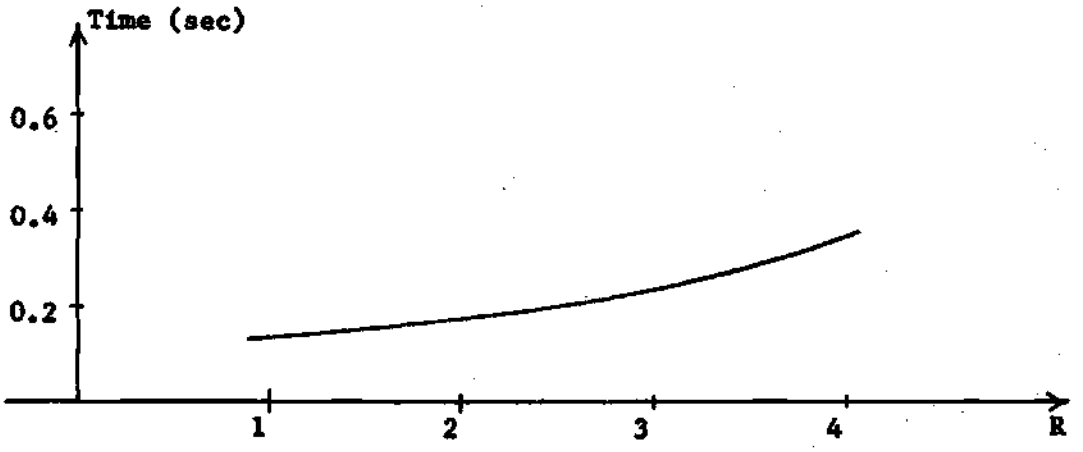


Figure 37. Average Computation Time vs. R.

not linear with R, however, since the complexity of the optimization algorithm is not linear with R. Figure 37 (p. 82) illustrates these facts for this example with a sampling period of one second. If the controller is to operate in real time, the time required for the longest possible calculation must be less than the sampling period of the system.

In summary, this chapter has illustrated via two examples the applicability of the theory in Chapters II through IV. Although both examples were of second order to facilitate graphical illustration of the results, the theory is completely general and may be applied to any order system. Results were presented to indicate the degree of optimality of the algorithm. In the next chapter some conclusions will be drawn regarding the possible use of the proposed controller as an on-line device.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

This dissertation considers the problem of the optimal control of a discrete-input system with constraints on the state variables and on the control variables. A solution algorithm is presented for the particular problem consisting of a linear system with quadratic cost and with a convex constraint on the states of the system and a magnitude constraint on the controls.

Conclusions

The problem of a discrete system with constraints on the state variables and on the control variables has received very little attention in the literature. Most of the proposed solutions presented to date involve the solution of a two-point boundary value problem which results when the constraints of the problem are adjoined to the functional to be extremized. All of the proposed solutions require long computation times which prohibits their use in an on-line controller. The principal contribution of this dissertation is an algorithm which does not involve the solution of a two-point boundary value problem and is capable of yielding a solution in real time.

The proposed controller consists of an R-step optimization scheme which gives an optimal solution when considering only R steps into the future. When the controls to be applied to the system are computed by

optimizing the cost functional over R steps, there exists the possibility that the computed control might force the state of the system to a point from which the original state boundary cannot be avoided with the control available. This problem would not exist with an infinite-step controller or one that gives a complete solution with one calculation. The concept of an additional boundary in the state space, referred to as the saturation boundary, is introduced to provide a means of handling this problem. The saturation boundary is the locus of all points in the state space from which a violation of the original convex state boundary can just be avoided with the control available.

Theorems I and II establish that any point on a trajectory, computed by the R-step optimal control scheme, which violates either the saturation boundary or the original boundary, should be bounded to the state boundary. A procedure is presented for bounding points on a trajectory to the state boundary. An explanation of the R-step controller is given and the method of extending the controller to any value of R is presented in Appendix I.

The principal limitation of the R-step control scheme presented in this dissertation is that it yields a suboptimal solution over the complete solution time interval $[t_0,T]$. This solution can be forced to approximate the optimal solution by the proper choice of R and sampling period of the system. Since the applied controls are constant over each sampling period, the total cost may be reduced by increasing the sampling rate thereby reducing the sampling period as explained in Chapter V. In the limit, the cost approaches the cost of the continuous system in which the controls are continuously variable. Practically, the

sampling rate cannot be increased without limit because of the finite computation time required during each sampling period. In addition, a large number of discrete systems have sampling rates fixed by other considerations than those listed above.

The other factors influencing the degree of optimality of the solution is the lead time used in each computation. The lead time should be made large enough so that the cost of each step cannot be reduced significantly by using a longer lead time. Since lead time is equal to R times the sampling period, it may be increased by increasing R. As R is increased, however, the complexity of the controller increases, resulting in a longer computation time. In order for the controller to operate in real time, each computation must be made during one sampling period. Therefore, eventually there has to be a trade-off between R and the sampling period. The ability to make this trade-off in order to obtain a real time controller, however, is precisely the advantage of the algorithm presented in this dissertation over all others presented in the past.

Recommendations

The optimization algorithm presented in this dissertation applies only to the control of a linear discrete-input system with quadratic cost and with convex constraints on the state variables and magnitude constraints on the control variables. An obvious extension of this work is the control of the same system with concave constraints on the states. A theorem similar to Theorem II in this dissertation would be needed to establish the necessary conditions for optimality. The results of this

dissertation can be extended to systems with non-quadratic cost functionals. Since both of the above extensions lead to the possibility of obtaining local minima as well as a global minimum, a more elaborate iterative search technique would be required.

The algorithm as presented in this dissertation solves the regulator problem. It is felt that the corresponding tracking problem can be handled with a few modifications to the proposed algorithm. The tracking problem, of course, involves a non-stationary state boundary as opposed to the stationary boundary used in the regulator problem. No attempt is made in this dissertation to improve the computation speed of the algorithm presented. Additional work on improving the speed of computation and in designing a special purpose digital controller would be useful.

As pointed out in Chapter V, a more optimal solution can be obtained by increasing R which leads to a computation time problem when bounding points to the state boundary. Since the controls with the states unconstrained are pre-computed as a function of the initial state $\underline{x}(0)$, the total cost may be reduced by computing these unbounded controls using a very large lead time (very large R) without increasing the computational time of the algorithm. This results in a controller that has two values of R; one for the unbounded-trajectory procedure and one for the bounded-trajectory procedure. This idea can be extended to a controller with a variable R which depends on the number of points and the particular points that are bounded.

APPENDICES

APPENDIX I

EXTENSION OF OPTIMIZATION ALGORITHM

The basic theory of the optimization algorithm was given in Chapters III and IV. The purpose of this appendix is to illustrate how the basic theory is implemented in an R-step controller for an nth order plant with m control variables where R is any positive integer. The general procedure used by a controller for any value of R is as follows:

(1) Computation of unconstrained controls.

(2) Computation of magnitude constrained controls.

(3) Computation of trajectory (states unconstrained).

(4) Determination if any points on trajectory in (3) lie above the state boundary.

(5) Recomputation of trajectory with points found to lie in an unacceptable region bounded to the state boundary.

Parts (1), (2), and (3) are the same for all values of R. The complexity of parts (4) and (5) increases as R is increased because of the larger number of points that have to be checked and possibly bounded.

The approach taken is to show how the optimization algorithm for R=2 may be obtained from the R=1 algorithm; how the R=3 algorithm may be obtained from the R=2 algorithm; how the R=4 algorithm may be obtained from the R=3 algorithm; and finally how the R=N+1 algorithm is obtained from the R=N algorithm.

Consider first the algorithm for the case where R=1 as shown in Figure 38. The second block in the optimization algorithm computes the unbounded optimal controls and the bounded optimal controls if any of the unconstrained controls exceed their bound. In addition, the trajectory using these control vectors is found. In the third block the Rth point is tested to determine its location with respect to the saturation boundary. If $\underline{x}(R)$ is below the S.B., then since R=1, the solution is acceptable and the next iteration is computed. If $\underline{x}(R)$ is not below the S.B., it is tested to determine if it lies to the right of the point of tangency (P.T.) and below the original linear boundary (O.L.B.). If so, this is an acceptable point as shown in Figure 39. If $\underline{x}(R)$ fails this test, it lies either to the left of the P.T. and above the S.B. or to the right of the P.T. and above the O.L.B. In either case, $\underline{x}(R)$ must be constrained to lie on the state boundary as shown in the last block in Figure 38.

Figure 40 shows the optimization algorithm for R=2. Note that the five blocks at the top of the flow diagram are the same as for the case for R=1 in Figure 38. In fact, these five blocks are the same regardless of the value of R. For the case where R=2, there are two points that have to be checked for acceptability. In block number three, if $\underline{x}(R)$ checks out to be below the S.B., then $\underline{x}(R-1)$ or $\underline{x}(1)$ must be below the S.B. since if it were not, then $\underline{u}(2)$ would have to be greater than its magnitude bound. But none of the control vectors computed in block number two exceeds its bound. Therefore, if $\underline{x}(R)$ passes the test in block three, all points on the trajectory are acceptable

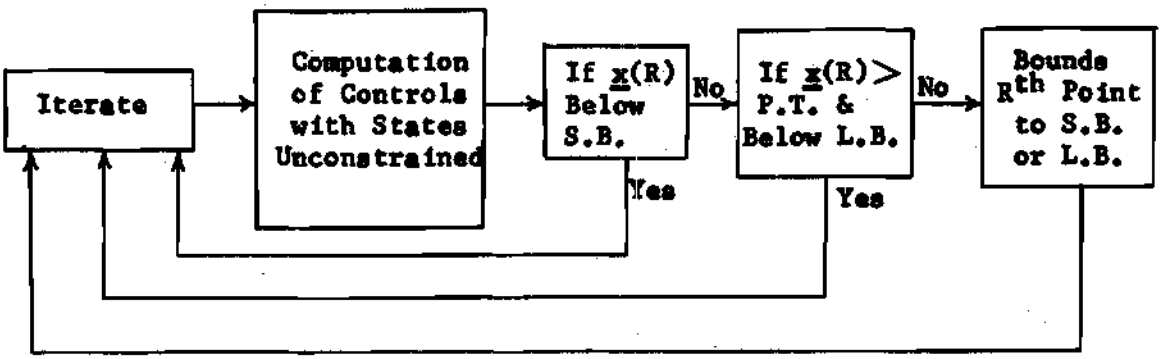


Figure 38. Optimization Algorithm for R = 1.

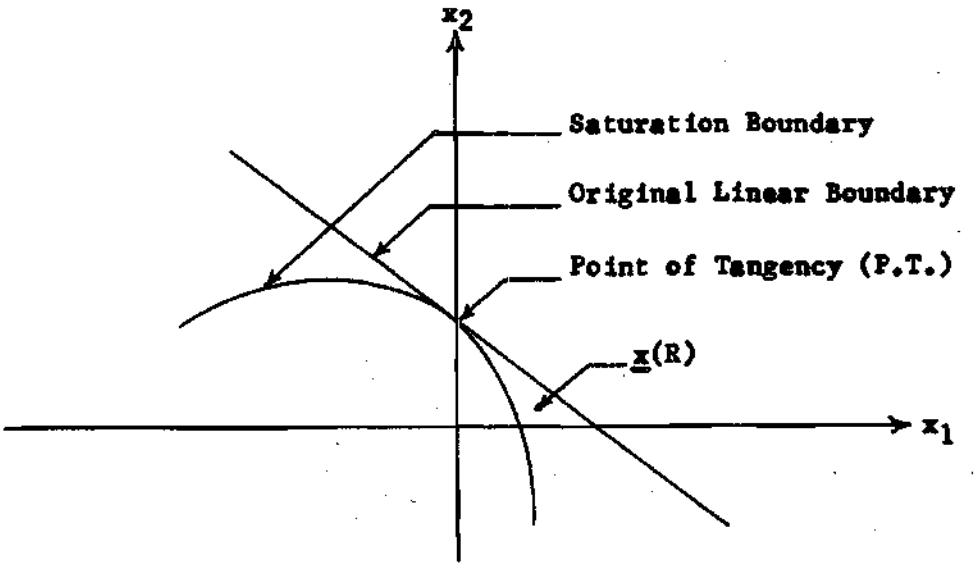


Figure 39. Point $\underline{x}(R)$ In Allowable Region.

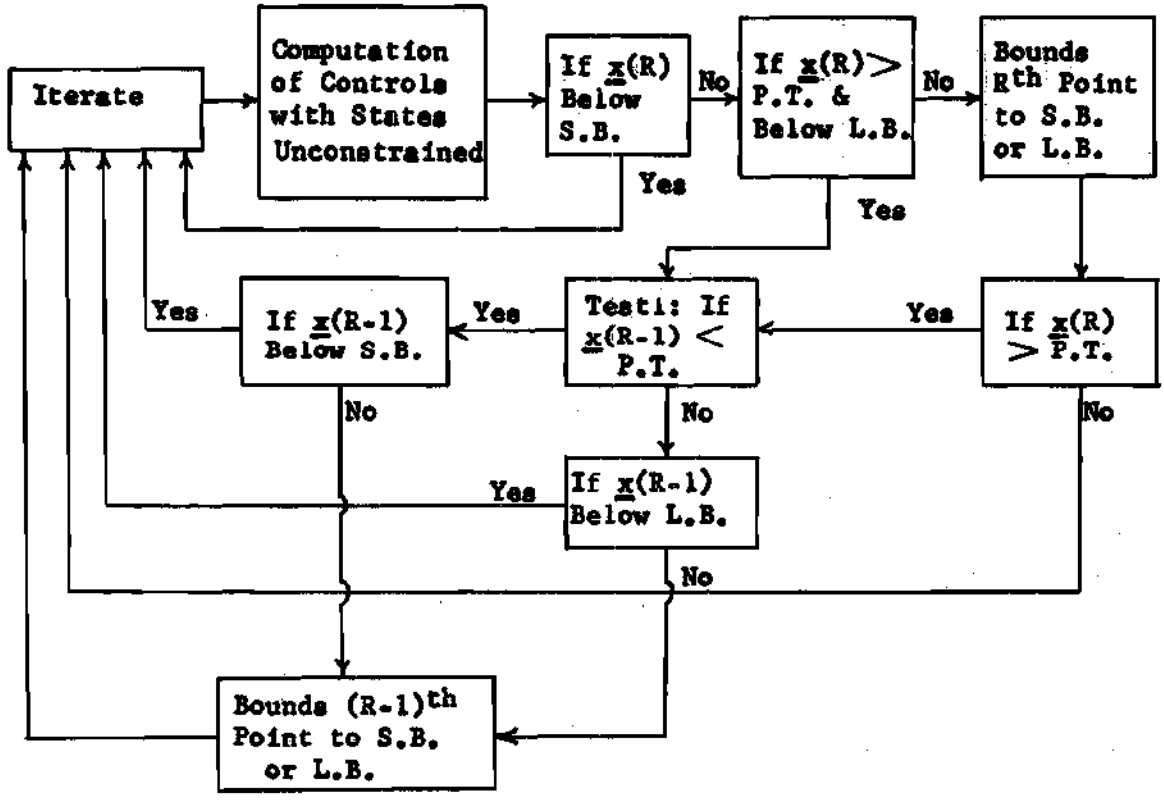


Figure 40. Optimization Algorithm for R = 2.

and the next iteration may be computed. If $\underline{x}(R)$ passes the test in block four, the point $\underline{x}(R-1)$ must be checked. Test 1 determines the location of $\underline{x}(R-1)$ with respect to the P.T. of the S.B. with the original linear boundary. If $\underline{x}(R-1)$ is to the left of the P.T. and below the S.B. or to the right of the P.T. and below the O.L.B., then it is an acceptable point and the next iteration may be computed. If $\underline{x}(R-1)$ fails either of these tests, the point $\underline{x}(R-1)$ must be bounded to the state boundary. If $\underline{x}(R)$ fails the test in block four, then the Rth point must be bounded to the state boundary in block five. The control components computed in the optimization procedure that bound $\underline{x}(R)$ to the state boundary are admissible controls. Therefore, if $\underline{x}(R)$ is to the left of the P.T. and on the S.B., then the point $\underline{x}(R-1)$ is acceptable and the next iteration may be computed. If $\underline{x}(R)$ is to the right of the P.T. and on the O.L.B., then the point $\underline{x}(R-1)$ must be checked as shown in Figure 40.

The flow diagram for R=2 is identical to the flow diagram for R=1 with additional blocks required to check the point $\underline{x}(R-1)$. The flow diagram for R=3 is shown in Figure 41. This flow diagram is identical to the flow diagram for R=2 in Figure 40 with additional blocks to test the point $\underline{x}(R-2)$ and additional blocks shown in the dotted section to bound the point $\underline{x}(R-1)$ to the state boundary. The flow diagram for R=4 is shown in Figure 42. This flow diagram is identical to the flow diagram for R=3 in Figure 41 with additional blocks to test the point $\underline{x}(R-3)$ and additional blocks shown in the dotted section to bound the point $\underline{x}(R-2)$ to the state boundary. The procedure of adding two sets of additional blocks to the algorithm for

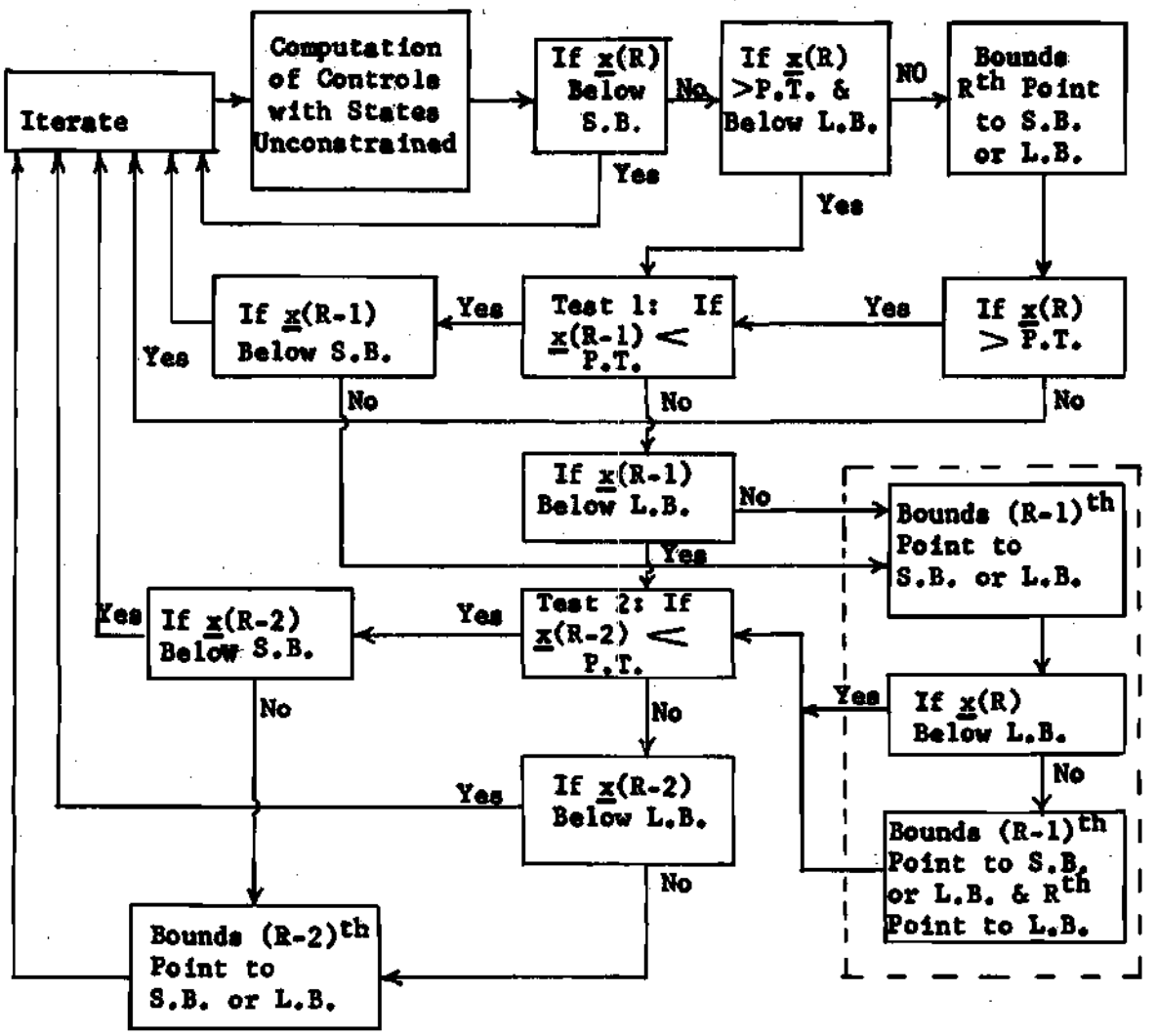


Figure 41. Optimization Algorithm for R = 3.

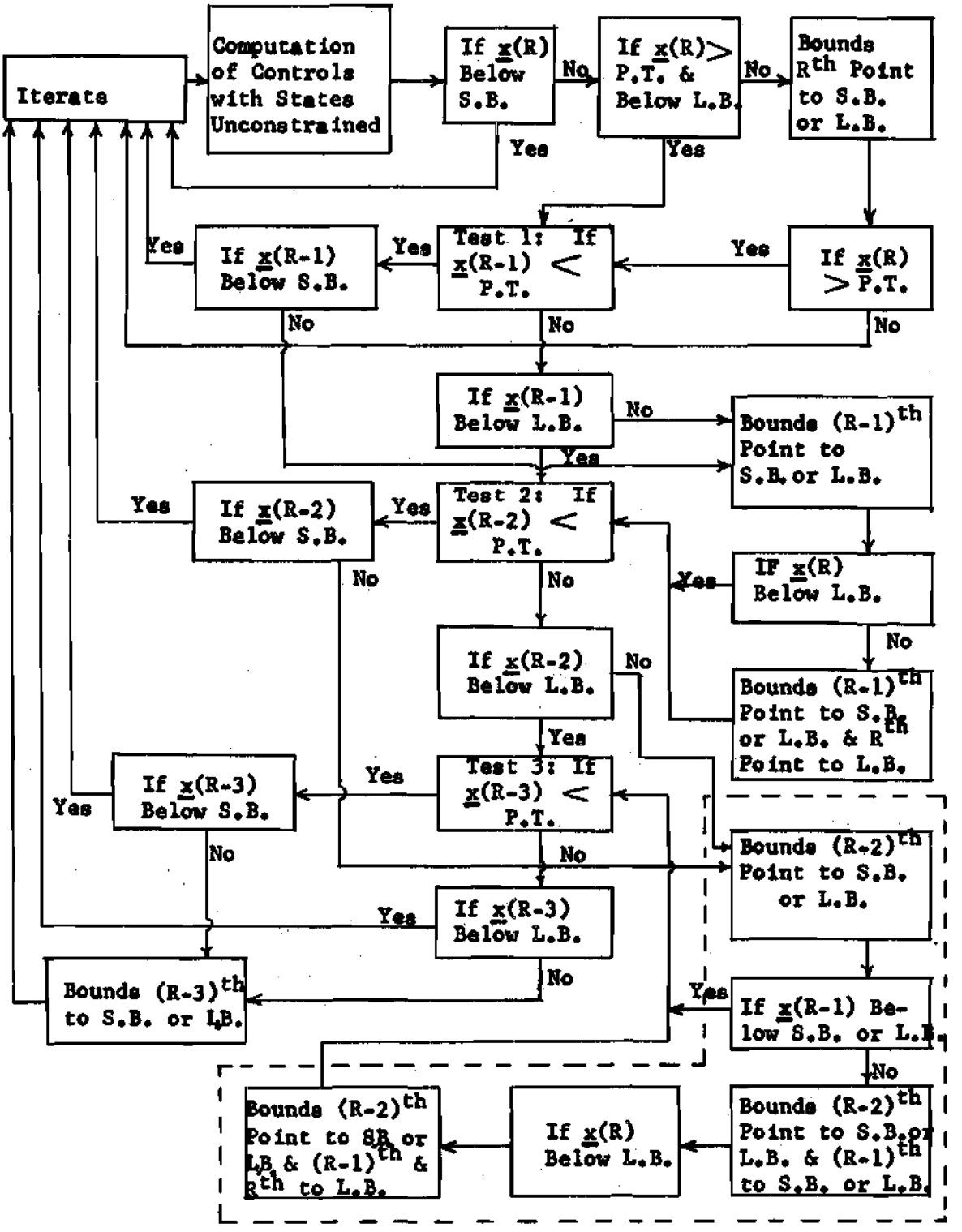


Figure 42. Optimization Algorithm for R = 4.

R=N to obtain the algorithm for R=N+1 is thus established.

APPENDIX II

CONTROL PROCEDURE FOR R=1 THROUGH R=4 WITH STATES UNCONSTRAINED

The results shown in the following table pertain to the system shown in Figure 5 with n state variables and m control variables. The table gives the specific expressions for R=1 through R=4 for the following:

(1) The cost functional in Equation (4.1).

(2) The definitions of the constant matrices in the cost functional in Equations (4.4), (4.6), and (4.8).

(3) Expressions for the unconstrained controls in Equation (3.14).

(4) Expressions for the magnitude constrained controls in the coordinatewise gradient procedure in Equations (3.15), (3.16), and (3.17).

The constant K matrices in the cost expressions are not defined since they do not appear in the control variable expressions.

Table 3. Cost Functionals and Control Variable Expressions in Optimization Algorithm for R=1 Through R=4

R = 1

$$J = 1/2\,[\underline{u}(1)^T C_1\,\underline{u}(1) + 2\underline{x}(0)^T C_2\,\underline{u}(1) + K_1]$$

$$C_1 = Q + G^T P G \qquad C_2 = F^T P G$$

$\underline{u}(1)^{(n+1)}$ - No iterative procedure necessary.

R = 2

$$J = 1/2\,[\underline{u}(1)^T C_1\,\underline{u}(1) + \underline{u}(2)^T C_2\,\underline{u}(2) + 2\underline{u}(1)^T C_3\,\underline{u}(2) + 2\underline{x}(0)^T C_4\,\underline{u}(1) + 2\underline{x}(0)^T C_5\,\underline{u}(2) + K_2]$$

$$C_1 = Q + G^T P G + G^T F^T P F G \qquad C_2 = Q + G^T P G$$

$$C_3 = G^T F^T P G \qquad C_4 = F^T P G + F^T F^T P F G \qquad C_5 = F^T F^T P G$$

$$\underline{u}(1) = (C_1 C_2 - C_3^2)^{-1}\,\underline{x}(0)^T\,(C_3 C_5 - C_2 C_4)$$

$$\underline{u}(2) = (C_3^2 - C_1 C_2)^{-1}\,\underline{x}(0)^T\,(C_1 C_5 - C_3 C_4)$$

$$\underline{u}(1)^{(n+1)} = -(C_1)^{-1}\,(C_3\underline{u}(2)^{(n)} + C_4)$$

$$\underline{u}(2)^{(n+1)} = -(C_2)^{-1}\,(C_3\underline{u}(1)^{(n)} + C_5)$$

R = 3

$$J = 1/2\,[\underline{u}(1)^T C_1\,\underline{u}(1) + \underline{u}(2)^T C_2\,\underline{u}(2) + \underline{u}(3)^T C_3\,\underline{u}(3) + 2\underline{u}(1)^T C_4\,\underline{u}(2) + 2\underline{u}(1)^T C_5\,\underline{u}(3) + 2\underline{u}(2)^T C_6\,\underline{u}(3) + 2\underline{x}(0)^T C_7\,\underline{u}(1) + 2\underline{x}(0)^T C_8\,\underline{u}(2) + 2\underline{x}(0)^T C_9\,\underline{u}(3) + K_3]$$

$$C_1 = Q + G^T P F + G^T F^T P F G + G^T F^T F^T P F F G$$

$$C_2 = Q + G^T P G + G^T F^T P F G \qquad C_3 = Q + G^T P G$$

$$C_4 = G^T F^T PG + G^T F^T F^T PFG \qquad C_5 = G^T F^T F^T PG$$

$$C_6 = G^T F^T PG \qquad C_7 = F^T PG + F^T F^T PFG + F^T F^T F^T PFFG$$

$$C_8 = F^T F^T PG + F^T F^T F^T PFG \qquad C_9 = F^T F^T F^T PG$$

$$D = (C_2C_6 - C_4C_5)C_6 + (C_1C_5 - C_4C_6)C_5 + (C_4^2 - C_1C_2)C_3$$

$$\underline{u}(1) = -D^{-1}\underline{x}(0)^T [(C_5^2 - C_2C_3)C_7 + (C_3C_4 - C_5C_6)C_8 + (C_2C_6 - C_4C_5)C_9]$$

$$\underline{u}(2) = -D^{-1}\underline{x}(0)^T [(C_3C_4 - C_5C_6)C_7 + (C_6^2 - C_1C_3)C_8 + (C_1C_5 - C_4C_6)C_9]$$

$$\underline{u}(3) = -D^{-1}\underline{x}(0)^T [(C_2C_6 - C_4C_5)C_7 + (C_1C_5 - C_4C_6)C_8 + (C_4^2 - C_1C_2)C_9]$$

$$\underline{u}(1)^{(n+1)} = -(C_1)^{-1} [C_4\underline{u}(2)^{(n)} + C_6\underline{u}(3)^{(n)} + \underline{x}(0)^T C_7]$$

$$\underline{u}(2)^{(n+1)} = -(C_2)^{-1} [C_4\underline{u}(1)^{(n)} + C_5\underline{u}(3)^{(n)} + \underline{x}(0)^T C_8]$$

$$\underline{u}(3)^{(n+1)} = -(C_3)^{-1} [C_6\underline{u}(1)^{(n)} + C_5\underline{u}(2)^{(n)} + \underline{x}(0)^T C_9]$$

R = 4

$$\begin{aligned} J = 1/2 [&\underline{u}(1)^T C_1\underline{u}(1) + \underline{u}(2)^T C_2\underline{u}(2) + \underline{u}(3)^T C_3\underline{u}(3) + \underline{u}(4)^T C_4\underline{u}(4) \\ &+ 2\underline{u}(1)^T C_5\underline{u}(2) + 2\underline{u}(1)^T C_6\underline{u}(3) + 2\underline{u}(1)^T C_7\underline{u}(4) \\ &+ 2\underline{u}(2)^T C_8\underline{u}(3) + 2\underline{u}(2)^T C_9\underline{u}(4) + 2\underline{u}(3)^T C_{10}\underline{u}(4) \\ &+ 2\underline{x}(0)^T C_{11}\underline{u}(1) + 2\underline{x}(0)^T C_{12}\underline{u}(2) + 2\underline{x}(0)^T C_{13}\underline{u}(3) \\ &+ 2\underline{x}(0)^T C_{14}\underline{u}(4) + K_4] \end{aligned}$$

$$C_1 = Q + G^T PG + G^T F^T PFG + G^T F^T F^T PFFG + G^T F^T F^T F^T PFFFG$$

$$C_2 = Q + G^T PG + G^T F^T PFG + G^T F^T F^T PFFG$$

$$C_3 = Q + G^T PG + G^T F^T PFG$$

$$C_4 = Q + G^T PG$$

$$C_5 = G^T F^T PG + G^T F^T F^T PFG + G^T F^T F^T F^T PFFG$$

$$C_6 = G^T F^T F^T PG + G^T F^T F^T F^T PFG$$

$$C_7 = G^T F^T F^T F^T PG \qquad C_8 = G^T F^T PG + G^T F^T F^T PFG$$

$$C_9 = G^T F^T F^T PG \qquad C_{10} = G^T F^T PG$$

$$C_{11} = F^T PG + F^T F^T PFG + F^T F^T F^T PFFG + F^T F^T F^T F^T PFFFG$$

$$C_{12} = F^T F^T PG + F^T F^T F^T PFG + F^T F^T F^T F^T PFFG$$

$$C_{13} = F^T F^T F^T PG + F^T F^T F^T F^T PFG$$

$$C_{14} = F^T F^T F^T F^T PG$$

$$N1 = C_2C_3C_4 + 2C_8C_9C_{10} - C_3C_9^2 - C_2C_{10}^2 - C_4C_8^2$$

$$N2 = C_3C_4C_5 + C_6C_9C_{10} + C_7C_8C_{10} - C_3C_7C_9 - C_5C_{10}^2 - C_4C_6C_8$$

$$N3 = C_4C_5C_8 + C_2C_7C_{10} + C_6C_9^2 - C_7C_8C_9 - C_2C_4C_6 - C_5C_9C_{10}$$

$$N4 = C_5C_8C_{10} + C_2C_3C_7 + C_6C_8C_9 - C_7C_8^2 - C_2C_6C_{10} - C_3C_5C_9$$

$$N5 = C_1C_3C_4 + 2C_6C_7C_{10} - C_3C_7^2 - C_1C_{10}^2 - C_4C_6^2$$

$$N6 = C_1C_4C_8 + C_5C_7C_{10} + C_6C_7C_9 - C_8C_7^2 - C_1C_9C_{10} - C_4C_5C_6$$

$$N7 = C_1C_8C_{10} + C_3C_5C_7 + C_9C_6^2 - C_6C_7C_8 - C_1C_3C_9 - C_5C_6C_{10}$$

$$N8 = C_1C_2C_4 + 2C_5C_7C_9 - C_2C_7^2 - C_1C_9^2 - C_4C_5^2$$

$$N9 = C_1C_2C_{10} + C_5C_7C_8 + C_5C_6C_9 - C_2C_6C_7 - C_1C_8C_9 - C_{10}C_5^2$$

$$N10 = C_1C_2C_3 + 2C_5C_6C_8 - C_2C_6^2 - C_1C_8^2 - C_3C_5^2$$

$$D = -(C_1N1 - C_5N2 + C_6N3 - C_7N4)$$

$$\underline{u}(1) = D^{-1}\underline{x}(0)^T [(C_{11}N1 - C_{12}N2 + C_{13}N3 - C_{14}N4)]$$

$$\underline{u}(2) = D^{-1}\underline{x}(0)^T [(-C_{11}N2 + C_{12}N5 - C_{13}N6 + C_{14}N7)]$$

$$\underline{u}(3) = D^{-1}\underline{x}(0)^T [(C_{11}N3 - C_{12}N6 + C_{13}N8 - C_{14}N9)]$$

$$u(4) = D^{-1}\underline{x}(0)^T [(-C_{11}N4 + C_{12}N7 - C_{13}N9 + C_{14}N10)]$$

$$\underline{u}(1)^{(n+1)} = -(C_1)^{-1} [C_5\underline{u}(2)^{(n)} + C_6\underline{u}(3)^{(n)} + C_7\underline{u}(4)^{(n)} + \underline{x}(0)^T C_{11}]$$

$$\underline{u}(2)^{(n+1)} = -(C_2)^{-1} [C_5\underline{u}(1)^{(n)} + C_8\underline{u}(3)^{(n)} + C_9\underline{u}(4)^{(n)} + \underline{x}(0)^T C_{12}]$$

$$\underline{u}(3)^{(n+1)} = -(C_3)^{-1} [C_6\underline{u}(1)^{(n)} + C_8\underline{u}(2)^{(n)} + C_{10}\underline{u}(4)^{(n)} + \underline{x}(0)^T C_{13}]$$

$$\underline{u}(4)^{(n+1)} = -(C_4)^{-1} [C_7\underline{u}(1)^{(n)} + C_9\underline{u}(2)^{(n)} + C_{10}\underline{u}(3)^{(n)} + \underline{x}(0)^T C_{14}]$$

BIBLIOGRAPHY

1. Pontryagin, Boltyanskii, Gamkrelidze, Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience Publishers, 1962.

2. R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, N.J., 1957.

3. S. G. Mikhlin, *Variational Methods in Mathematical Physics*, The MacMillan Company, Oxford, England, 1964.

4. Julius Tou, *Modern Control Theory*, McGraw-Hill, New York, N.Y., 1964.

5. C. A. Desoer and J. Wing, "The Minimal Time Regulator Problem for Linear Sampled-Data Systems: General Theory," *Journal of Franklin Institute*, Sept., 1961.

6. J. L. LeMay, "Recoverable and Reachable Zones for Control Systems with Linear Plants and Bounded Controller Outputs," *IEEE Transactions on Auto Control*, Oct., 1964.

7. A. Nagata, S. Kodama, and S. Kumagai, "Time Optimal Discrete Control System with Bounded State Variables," *IEEE Transactions on Auto Control*, April, 1965.

8. P. E. Sarachik and G. M. Kranc, "Optimal Control of Discrete Systems with Constrained Inputs," *Journal of Franklin Institute*, Jan., 1964.

9. S. E. Dreyfus, "Variational Problems with Inequality Constraints," *Journal of Mathematical Analysis and Applications*, Vol. 4, No. 2, April, 1962.

10. L. D. Berkovitz, "On Control Problems with Bounded State Variables," *Journal of Mathematical Analysis and Applications*, Vol. 5, No. 3, Dec., 1962.

11. Z. V. Rekasius, "Suboptimal Design of Intentionally Nonlinear Controllers," *IEEE Transactions on Auto Control*, Oct., 1964.

12. A. V. Fiacco and G. P. McCormick, "A Penalty Method for Optimal Control," *Proceeding of the IEEE*, Jan., 1967.

13. L. S. Lasdon, A. D. Waren and R. K. Rice, "An Interior Penalty Method for Inequality Constrained Optimal Control Problems," *IEEE Transactions on Auto Control*, Aug. 1967.

14. G. N. Saridis and Z. V. Rekasius, "Design of Approximately Optimal Feedback Controllers for Systems with Bounded States," *IEEE Transactions Auto Control,* Aug. 1967.

15. J. H. Schlag, "On the Optimal Control of Discrete Systems with Bounded State Variables," Ph.D. Thesis, Georgia Institute of Technology, Sept., 1967.

16. F. H. Kishi, "On Line Computer Control Techniques and Their Application to Re-entry Aerospace Vehicle Control," *Advances in Control Systems,* Vol. 1 (1964) Ed. by C. T. Leondes, Academic Press, Inc.

17. D. H. Chyung, "An Approximation to Bounded Phase Coordinate Control Problem for Linear Discrete Systems," *IEEE Transactions on Auto Control,* Feb., 1967.

18. Y. C. Ho and P. B. Brentani, "On Computing Optimal Control with Inequality Constraints," SIAM, 1963.

VITA

Joseph Samuel Boland, III, son of Joseph Samuel and Mary Authur (Owens) Boland, Jr., was born in Montgomery, Alabama on September 23, 1939. He attended the Montgomery, Alabama public schools and was graduated from Sidney Lanier High School in May of 1957. He attended Auburn University and received his B.S.E.E. in August of 1961. Mr. Boland entered the Graduate School of Auburn University in September of 1961, where he held the position of Instructor until June of 1962. From July of 1962 until July of 1964 he served as a lieutenant in the United States Army. He then accepted a position as a Graduate Assistant with the Auburn Research Foundation and re-entered the Graduate School of Auburn University where he received his M.S.E.E. in August of 1965. In September of 1965 he entered the Graduate School of the Georgia Institute of Technology where he held the position of Instructor of Electrical Engineering until September of 1968.

In June of 1962, Mr. Boland married the former Myrrium Rowena Haynes, daughter of Morse and Beulah (Holmes) Haynes of LaFayette, Alabama. He will complete the requirements for his Ph.D. E.E. in the field of automatic control in September of 1968.